PUBLICACIONES
DE LA UNIVERSIDAD DE ALICANTE

INTRODUCCIÓN A TCP/IP

Sistemas de Transporte de Datos

Luis Miguel Crespo Martínez
Francisco A. Candelas Herías

TD
TEXTOS DOCENTES

Publicaciones de la Universidad de Alicante, 1998

Portada: Gabinete de Imagen y Comunicación Gráfica
Universidad de Alicante

ISBN: 84-7908-435-9

Depósito Legal: A-1394-1998

Fotocomposición:

Espagrafic

Imprime: INGRA Impresores

Edición electrónica:

Espagrafic

Índice

El objetivo fundamental del presente libro es proporcionar una visión adaptada a la realidad en cuanto al diseño e implementación de redes basadas en la pila de protocolos TCP/IP.

Para ello, se utiliza como soporte de trabajo una configuración típica en instalaciones industriales y empresa privada. Éste será el punto de partida para los numerosos ejemplos prácticos que vienen propuestos a lo largo de la obra.

Es conveniente, aunque no necesario, disponer de un mínimo de conocimientos referentes al sistema operativo Unix, así como nociones básicas en Redes de Computadores, puesto que de este modo se logrará una asimilación más sencilla a nivel conceptual y práctico.

Este libro, dadas sus características, es recomendable como manual de prácticas en estudios de Ingeniería Informática, Telecomunicación o Industrial.

En el primer capítulo se realiza una introducción a los niveles físicos y de enlace empleados en la configuración de referencia, es decir, Ethernet y Slip.

A continuación, se describen los aspectos más funcionales del protocolo de red IP, y su configuración sobre Unix y MSDOS. Se emplea un producto comercial denominado TUN, que permite la integración de aplicaciones basadas en TCP/IP entre ambas plataformas.

Por último, se describe el funcionamiento de los protocolos de transporte TCP y UDP para poder abordar, a continuación, algunas de las aplicaciones TCP/IP más usuales a nivel profesional, tales como Telnet, Snmp, Nfs y Ftp.

Para que el lector pueda evaluar el nivel de comprensión adquirido, se han intercalado a lo largo del libro una serie de cuestionarios con propuestas referentes a la materia expuesta.

Introducción a TCP/IP

Trataremos de tomar contacto con lo que hoy en día es el protocolo de red más extendido y con mayor aceptación a nivel mundial, es decir, TCP/IP.

Este protocolo que comenzó a finales de los años 60 como un proyecto de investigación financiado por el gobierno de EE.UU., sobre la conmutación de paquetes, se ha convertido en la década de los 90 en la estructura de red más ampliamente utilizada.

Se trata verdaderamente de un sistema abierto en la medida en que la definición de la serie de protocolos, así como muchas de sus implantaciones se encuentran disponibles al público, gratuitamente o a un módico precio. Ello constituye la base para lo que se conoce como la red **Internet mundial,** una red de larga distancia **(WAN)**, con más de 1 millón de ordenadores que literalmente invade el planeta.

Aunque actualmente y con motivo del creciente desarrollo tecnológico están surgiendo nuevos protocolos basados en tecnologías «Fast Paquet» (ATM, Frame Relay), TCP/IP se ha convertido en un estándar utilizado como protocolo de red nativo para sistemas Unix, estando sujeto de forma continua a nuevas implementaciones y mejoras.

Para abordar la materia, utilizaremos el sistema operativo Unix de SCO en los Servidores y MS-DOS en las estaciones

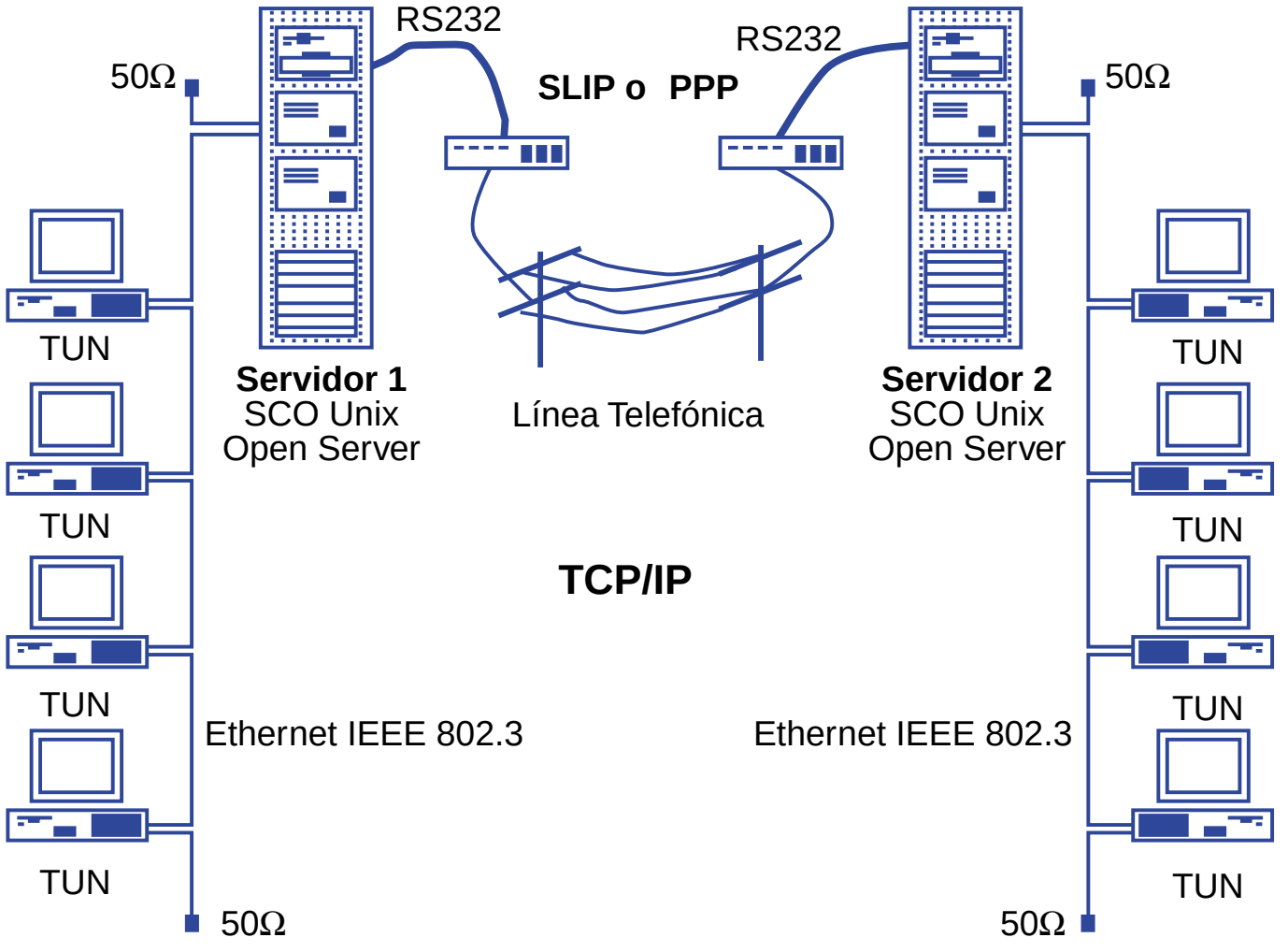


Figura 1. Topología de Referencia

de trabajo. La forma de integrar ambos sistemas operativos se realizará a través de un producto comercial denominado TUN, cuya finalidad es la de realizar las funciones de cliente en las conexiones TCP (o UDP) que se establezcan.

La topología de referencia sobre la que realizaremos nuestros ensayos, queda representada en la figura 1.

El objetivo de la presente ***documentación***, es proporcionar una visión general sobre la mecánica de funcionamiento de los distintos protocolos que conforman TCP/IP utilizando como soporte físico la encapsulación Ethernet.

2. Topología

Puesto que nos encontramos con una configuración compuesta de 2 segmentos en cable coaxial fino tipo 10Base2, sería conveniente comprobar algunas de las prescripciones definidas en la normativa IEEE802.3 (Todas, sería muy complicado), como lo son:

Impedancia característica del cable 50Ω +/- 2Ω

Atenuación del segmento: 8.5dB a 10Mhz (Oscilador 10Mhz y osciloscopio)

Velocidad de propagación mínima 0.65C (fabricante o Scaner)

Flanco de subida de la señal 25 +/- 5nS para pasar del 10% al 90%

* Resistencia en cortocircuito del cable R < 50mΩ /m
* Resistencia en cortocircuito del segmento

(Cable+conectores+malla)< 10Ω

* Resistencia de la interfaz de red sin alimentación R > 100 KΩ

Resistencia de la interfaz de red con alimentación R > 7.5 KΩ

Capacidad de entrada de la interfaz de red sin alimentación C < 6pF

* Resistencias de terminación de 50Ω +/- 2Ω

Longitud máxima del segmento de 185m

Las tarjetas de red utilizadas están provistas de dos tipos diferentes de interfaces Ethernet.

El utilizado en la mayoría de los casos es el tipo BNC para cable 10Base2. Esto quiere decir que la placa tiene integrados los dos módulos que conforman el nivel físico en las LAN: el **PSS** (Phisical Signaling Sublayer) o también conocido como C/D cuya principal finalidad es la codificación y decodificación de la señal (Manchester diferencial) así como la generación y detección de funciones especiales de control; y el **PMA** (Phisical Medium Attachement) o AM (Adaptación al medio) que tiene por objeto aplicar las señales eléctricas o

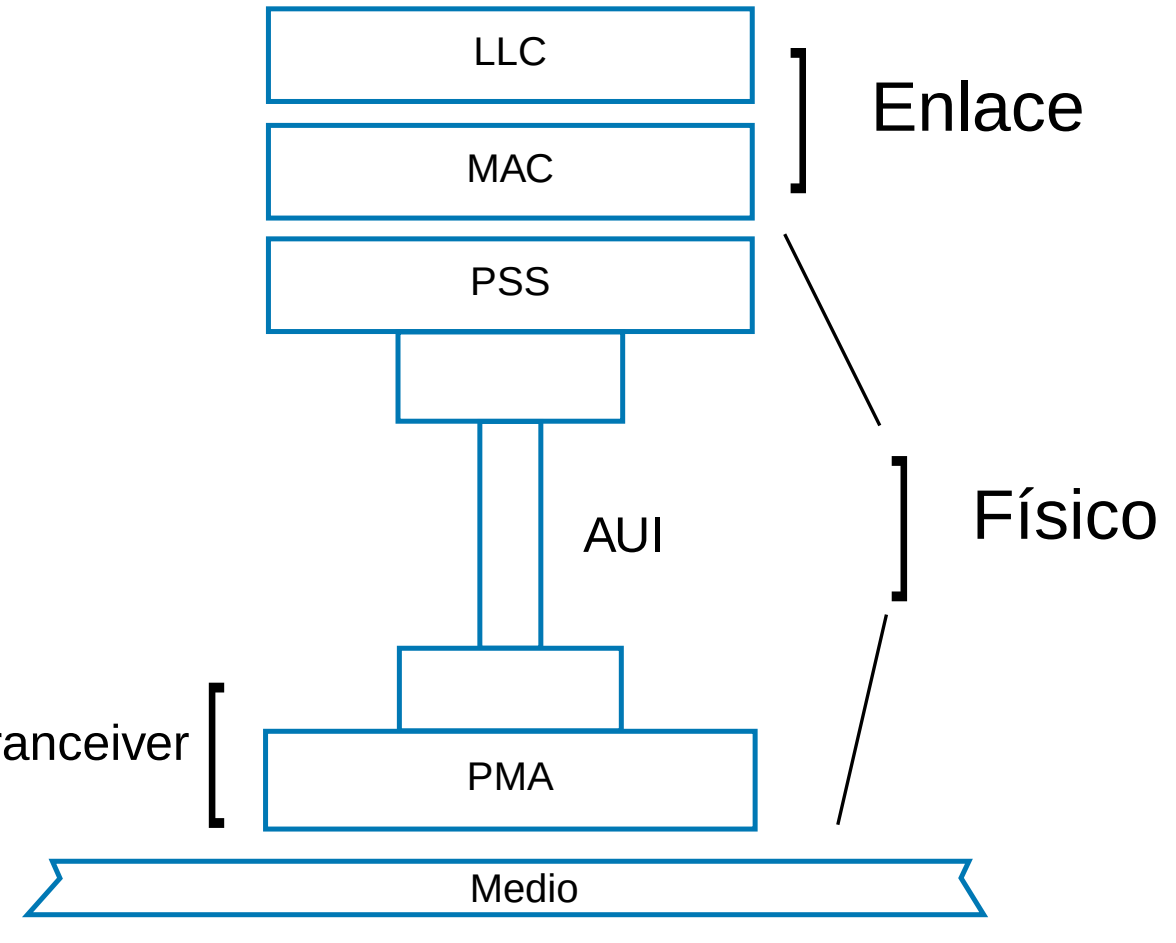
modular, según sea banda ancha o banda base, sobre el medio utilizado.

En nuestro caso, por tratarse de cable 10Base2, se trabaja directamente en banda base mediante una señal cuadrada con amplitudes del orden de 2.5 Voltios.

El PMA se encuentra conectado físicamente con el PSS a través de un tipo de interface denominado AUI, consistente en una serie de señales implementadas sobre controladores de línea en modo balanceado y con par trenzado en el caso de que la unión sea externa (Cable Drop). Esta conexión se externiza mediante un conector de 15 polos situado junto a la salida BNC de la tarjeta. (Figura 2)

La ventaja de disponer de este tipo de salida alternativa, es la flexibilidad que proporciona al poder elegir un **PMA** o **Transceptor** para el tipo de medio que queramos utilizar, ya sea fibra óptica, par trenzado, radioenlace, etc.

Lógicamente, para poder hacer uso de la salida **AUI** o la **BNC** se necesita indicar la opción deseada a la tarjeta, en nuestro caso por la activación de un pequeño puente situado sobre ella. En otro tipo de tarjetas esta tarea se realiza mediante un software suministrado por el fabricante.



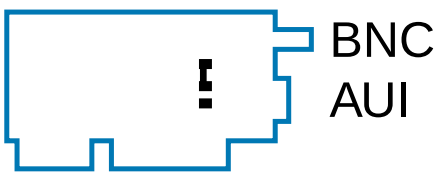


Figura 2. Estructura interna de la placa

Cuando se sobrepasa la longitud de un segmento, puede recurrirse al uso de repetidores. Estos dispositivos operan únicamente a nivel físico, propagando las señales a través de todos los segmentos conectados.

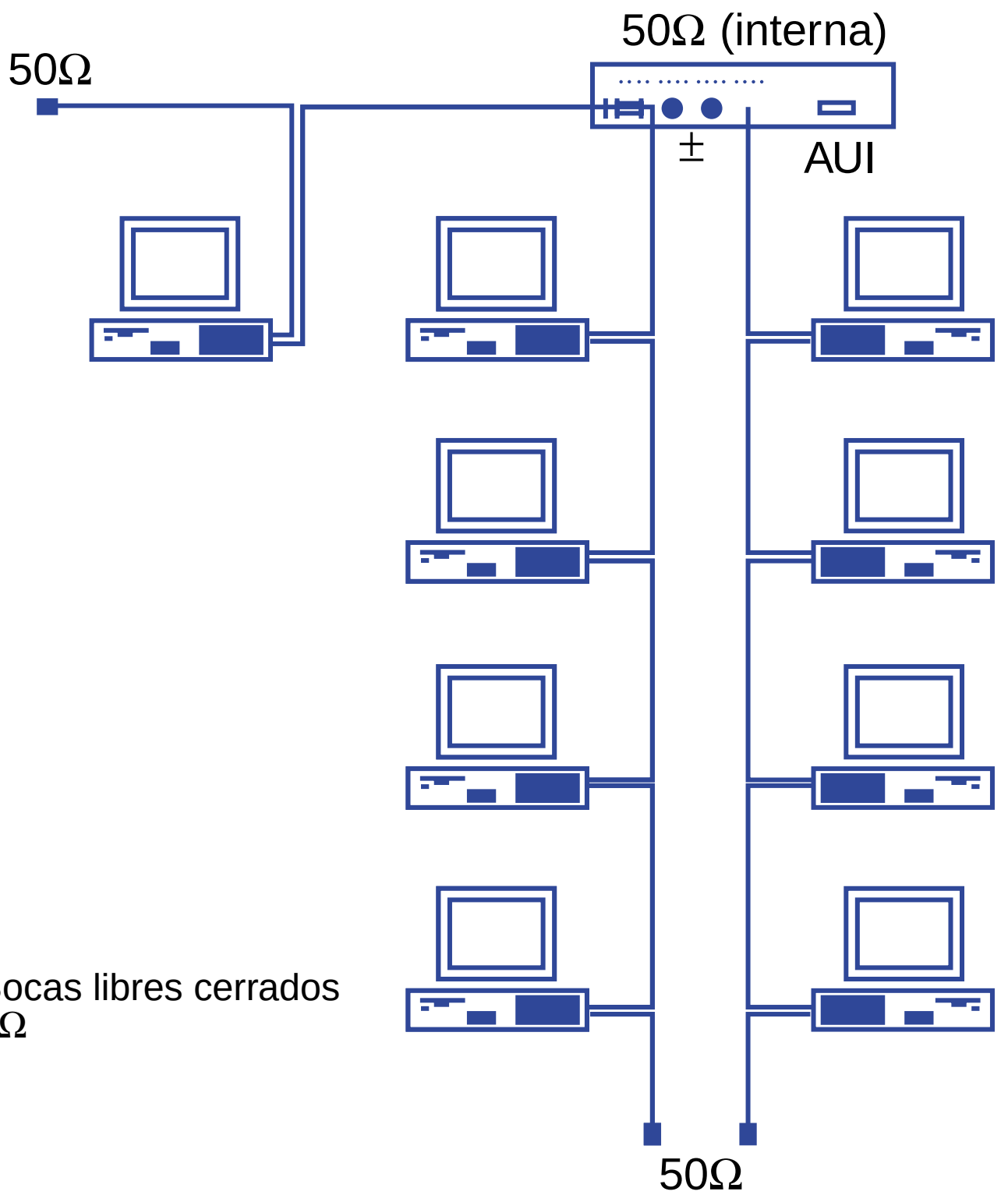
Un repetidor, es un dispositivo autónomo con alimentación de corriente propia y dotado de una serie de salidas de red que

pueden ser de tipo coaxial, par trenzado, fibra óptica o de tipo **AUI,** que como ya se ha explicado anteriormente, permite la conexión de **Transceptores** externos proporcionando una flexibilidad adicional en cuanto a conectividad se refiere.

Como era de esperar, los repetidores modernos ofrecen una serie de prestaciones adicionales que sus antecesores no poseían. Cabe destacar lo que se conoce como «particionamiento» consistente en la capacidad de detectar un exceso de colisiones o ruido en un segmento determinado y el aislamiento de ese segmento de forma automática, con el fin de evitar la propagación de señales nocivas hacia el resto de la red. (Figura 3)

Se debe tomar especial precaución con las salidas no utilizadas, puesto que si no se terminan correctamente con una resistencia, pueden generar unos niveles de ruido que serán repetidos hacia el resto de los segmentos (excepto en aquellos repetidores que dispongan de «particionamiento»), produciendo fallos de diversa índole en toda la red.

Según el tipo de fabricante, suele ser bastante común la incorporación de la resistencia de terminación dentro de la electrónica del repetidor, esto debe de ser comprobado antes de conectar el segmento. Normalmente existen unos peque-



* Nota: Bocas libres cerrados con 50Ω

Figura 3

ños puentes sobre la placa base del repetidor, cerca de cada boca de salida, que permiten la activación o desactivación de la resistencia de terminación.

2. Topología

Desde hace algunos años, la tendencia actual dentro de Ethernet, es la utilización del par trenzado como medio físico. Entre otras ventajas, presenta una mayor fiabilidad, un mayor control sobre las estaciones puesto que su topología aparente es en estrella y los concentradores de par trenzado o HUB UTP son casi en su totalidad gestionables. Esto significa que la localización de averías en el cableado es inmediata, cosa que no ocurre con 10Base2 o coaxial fino.

Concentrador de Par Trenzado (HUB)
Para IEEE 802.3

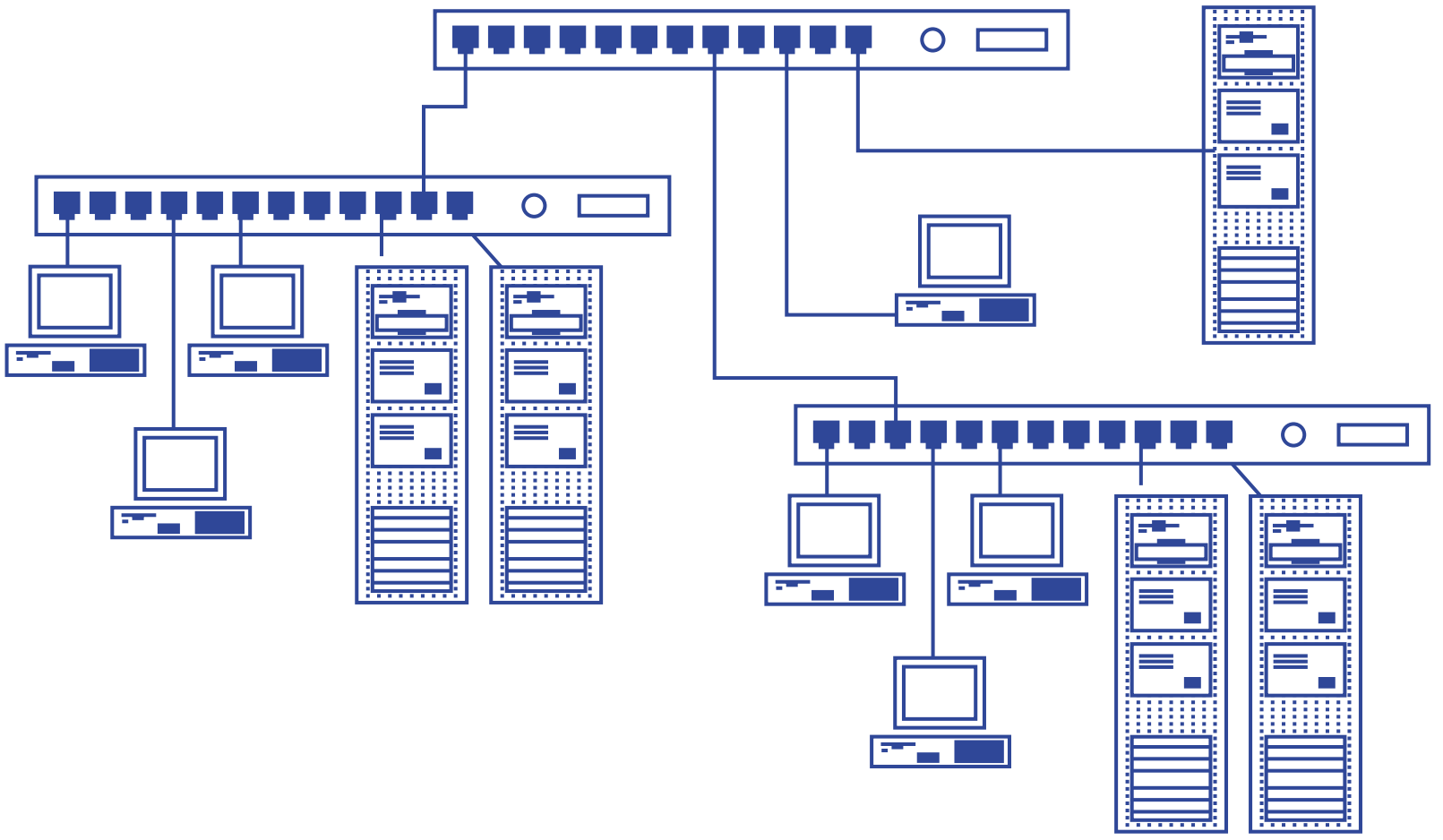


Figura 4

Una ventaja adicional del par trenzado, es la posibilidad de integrar el cableado de telefonía con el de datos, permitiendo al usuario mediante conmutaciones en el cuadro de distribución, la libre elección del tipo de toma que desea en su puesto de trabajo (telefonía o datos en varias combinaciones).

El cable en par trenzado utilizado, presenta un menor coste que el coaxial, aunque requiere una mayor tirada de líneas.

3. El primer protocolo

3.1 RFC

Antes de continuar, es necesario realizar un pequeño inciso. Todos los estándares oficiales de la comunidad Internet que como ya se ha comentado anteriormente se apoya directamente sobre TCP/IP, son publicados bajo la forma de **RFC** (Request for Comments). Además de ello, existen muchas RFC que no son estándares oficiales, pero que son publicadas con fines informativos.

Una RFC puede estar formada desde 1 a más de 200 páginas. Cada una se identifica por un número, como por ejemplo la RFC 1122, de forma que la más reciente tiene un número más alto.

Todas las RFC están disponibles gratuitamente a través de mensajerías electrónicas, o mediante FTP sobre Internet. El

envío de un correo electrónico como el siguiente, devuelve una lista detallada de las diferentes formas de obtener RFC:

To: rfc-info@ISI.EDU

Subject: gettings rfcs

help: ways_to_get_rfcs

3.2 Ethernet

El método de acceso utilizado por Ethernet es CSMA/CD, pero eso no es suficiente para que dos sistemas abiertos puedan coexistir en el mismo medio.

Existen básicamente dos tipos de tramas dentro de Ethernet. La denominada IEEE802.2/802.3 y la Ethernet. La utilización de una u otra depende del protocolo utilizado. Así, TCP/IP utiliza la encapsulación denominada Ethernet II, mientras que el protocolo utilizado por Novell, el IPX/SPX, utiliza la encapsulación 802.3. Ello deberá de ser tenido en cuenta si se desea hacer funcionar de forma simultánea Unix y Novell utilizando el mismo soporte físico.

En las estaciones de trabajo, se deberá escoger el controlador ODI si se desea hacer coexistir Novell y Unix. En caso contrario, es más efectivo utilizar Packet Driver. Estas opciones se encuentran disponibles en el menú de configuración

del programa TUNTCP.EXE. El controlador ODI, es capaz de simular dos tarjetas de red virtuales sobre una única tarjeta física; redireccionando así las tramas Ethernet al núcleo de TCP/IP y las 802.3 al núcleo de IPX.

3.2.1 Encapsulación en cola (Trailer Encapsulation)

La RFC 893 (1984) describía otra forma de trama utilizada por redes Ethernet, donde los campos de longitudes variables al principio del área de datos (Cabeceras IP y TCP) eran desplazados al final justo antes del CRC. Ello permitía a la porción de datos de la trama ser ubicada enteramente en una única página de memoria, ahorrando así una copia de memoria a memoria cada vez que los datos eran guardados en el espacio de direccionamiento del núcleo.

Encapsulación IEEE 802.2/802.3 (RFC 1042):

802.3 MAC			802.2 LLC			802.2 SNAP			
Dir. Destino	Dir. Fuente	Long.	DSAP	SSAP	Cntl.	org code	Tipo.	Datagrama IP	CRC
6	6	2	1	1	1	3	2	38 - 1492	4

Encapsulación Ethernet (RFC 894):

Dir. Destino	Dir. Fuente	Tipo.		Datagrama IP	CRC
6	6	2		46 - 1500	4

Figura 5

3.3 MTU

Como puede verse en la figura anterior, existe un límite en el tamaño de la trama tanto para Ethernet como para IEEE802.3. Los tamaños máximos para datos (datagrama IP) son de 1500 y 1492 respectivamente. Esta característica de la capa de enlace se conoce como MTU.

Si un datagrama IP es mayor que el MTU de la capa de enlace, IP utiliza un mecanismo denominado **fragmentación,** consistente en romper el datagrama en pequeños fragmentos de manera que cada uno de ellos sea menor que el MTU.

Este parámetro varía en función del tipo de enlace, y depende de diversos factores como la tasa de error media (BER), política de acceso al medio, velocidad de transmisión, etc.

3.4 Direcciones MAC

Cuando un datagrama es enviado hacia una estación o un servidor determinado, se necesita conocer ante todo su dirección física. Según la bibliografía consultada, puede aparecer también como dirección Ethernet, dirección MAC, etc.

Se trata de una combinación de 48 bits, de forma que los 3 primeros bytes de la izquierda corresponden al fabricante de la tarjeta, y los 3 siguientes dependen de diversos factores.

De aquí se deduce que utilizando los comandos adecuados, es fácil conocer la procedencia de cada tarjeta conectada a la red. La dirección MAC es única e irrepetible.

Toda tarjeta de red, debe ser capaz de responder a 2 direcciones: La suya propia, y la dirección de *Broadcast*. Esta última se caracteriza por tener los 48 bits a 1, con lo cual queda como FF:FF:FF:FF:FF:FF.

Una trama ethernet enviada a la dirección de *Broadcast* será atendida por todas las estaciones conectadas sobre el mismo segmento. Esta particularidad es utilizada ampliamente por el protocolo **ARP** que veremos más adelante.

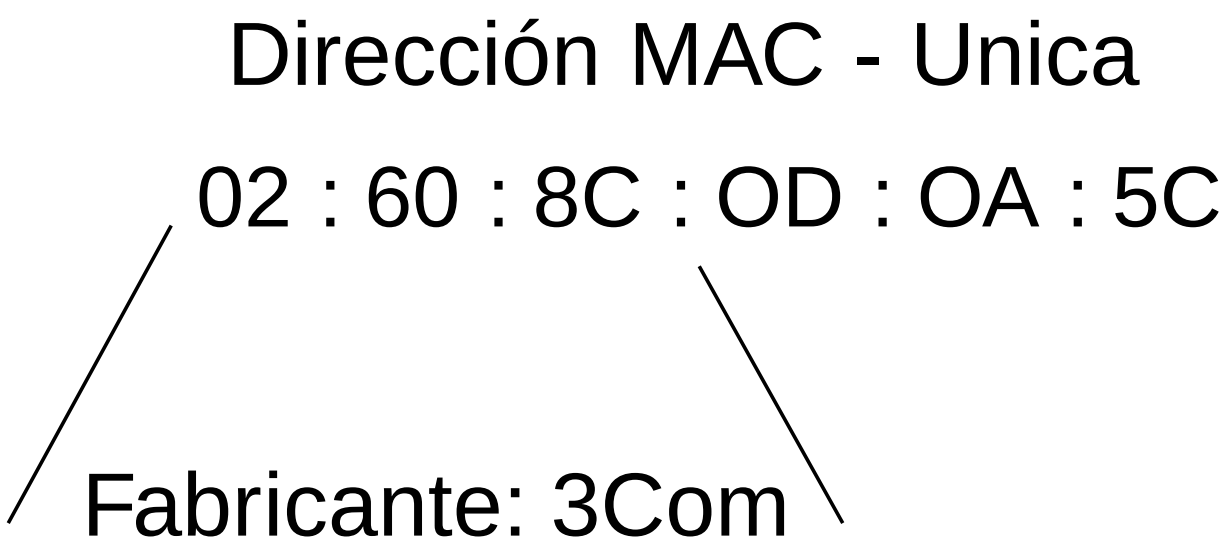


Figura 6

Una forma de saber si nuestro interface está correctamente instalado, es interrogar al hardware acerca de la dirección MAC. En Unix existen comandos para ello que luego veremos, y en MS-DOS se visualiza automáticamente al ejecutar el núcleo de TCP/IP, en nuestro caso el ejecutable ETHTCP.EXE de TUN mediante el fichero batch AUTOTCP.BAT situado en el directorio C:\TUNTCP.

Se detalla a continuación una lista con los principales fabricantes de dispositivos LAN Ethernet:

Dirección MAC	Fabricante	Dirección MAC	Fabricante
00:00:0C	Cisco	08:00:0B	Unisys
00:00:0F	NeXT	08:00:10	AT&T
00:00:10	Sytek	08:00:11	Tektronix
00:00:1D	Cabletron	08:00:14	Excelan
00:00:65	Network General	08:00:1ª	Data General
00:00:6B	MIPS	08:00:1B	Data General
00:00:77	MIPS	08:00:1E	Apollo
00:00:89	Cayman Systems	08:00:20	Sun
00:00:93	Proteon	08:00:25	CDC
00:00:A2	Wellfleet	08:00:2B	DEC
00:00:A7	NCD	08:00:38	Bull
00:00:A9	Network Systems	08:00:39	Spider Systems
00:00:C0	Western Digital	08:00:46	Sony
00:00:C9	Emulex	08:00:47	Sequent
00:80:2D	Xilogics Annex	08:00:5A	IBM
00:AA:00	Intel	08:00:69	Silicon Graphics
00:DD:00	Ungermann-Bass	08:00:6E	Excelan
00:DD:01	Ungermann-Bass	08:00:86	Imagen/QMS
02:07:01	MICOM/Interlan	08:00:87	Xyplex terminal
02:60:8C	3Com	08:00:89	Kinetics
00:C0:F0	Kingston	00:00:D0	D-LINK
08:00:02	3Com(Bridge)	08:00:8B	Pyramid
08:00:03	ACC	08:00:90	Retix
08:00:05	Symbolics	AA:00:03	DEC
08:00:08	BBN	AA:00:04	DEC
08:00:09	Hewlett-Packard	10:00:5A	IBM

Tabla 1

4. El protocolo IP

4.1 Direcciones IP

Las direcciones MAC permiten identificar máquinas dentro de un mismo segmento, pero ello no es suficiente para satisfacer las necesidades de comunicación dentro de una red que puede estar compuesta por miles de ellos. Se necesita pues un protocolo de red que permita hacer llegar a su destino una unidad de información, datagrama IP en nuestro caso, que a lo largo de su recorrido puede atravesar redes con protocolos de enlace muy dispares (Ethernet, Token Ring, Token Bus, líneas punto a punto con SLIP, PPP, HDLC y un sinfín de combinaciones a través de otras redes como RDSI o Frame Relay).

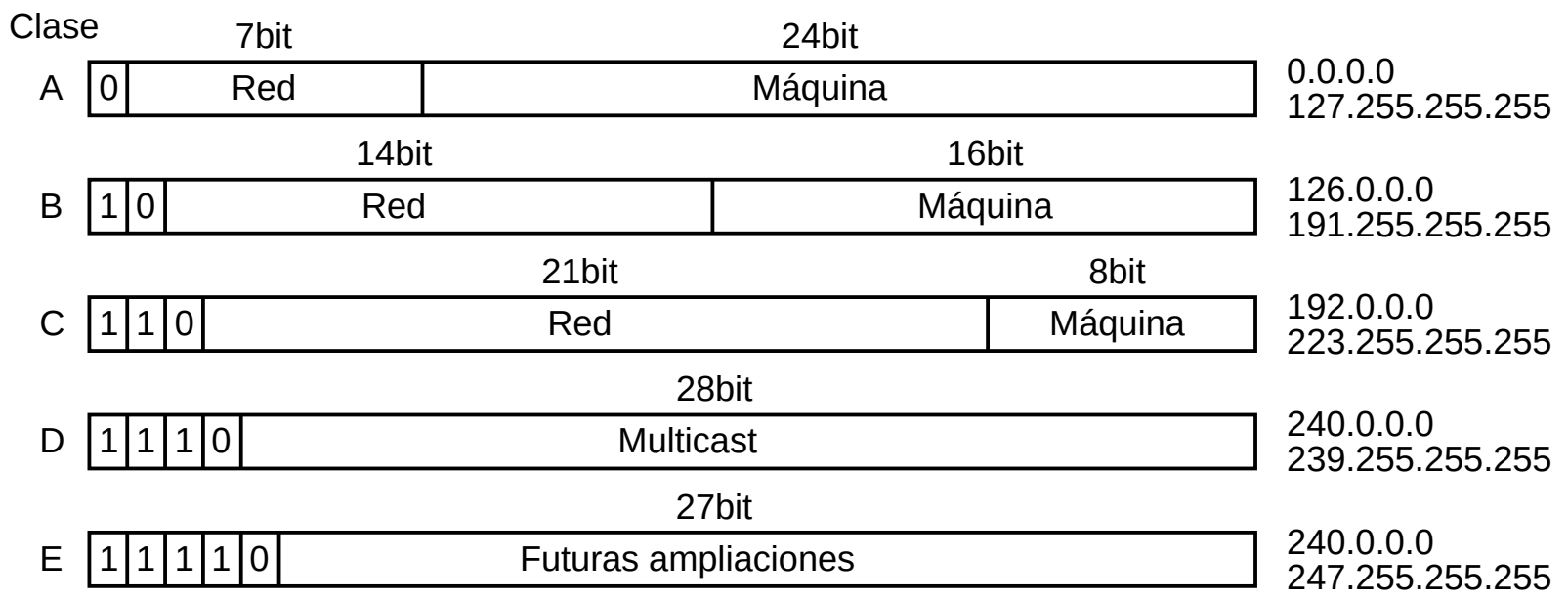
Las direcciones IP tienen una longitud de 32 bits, organizadas en 4 grupos de 8 bits cada uno. Se dividen fundamental-

mente en dos partes: la porción de la Red y la porción de la máquina.

La porción de red identifica a un grupo de máquinas que comparten el mismo protocolo de enlace dentro del mismo medio físico. En nuestra configuración de referencia, tenemos realmente tres redes: dos de tipo Ethernet y una de tipo punto a punto (Conexión RS232) en la que sólo intervienen dos partes correspondientes a cada extremo del enlace.

El campo de máquina hace referencia a todas aquellas estaciones conectadas a la misma red.

El tamaño de cada parte depende del valor de los bits de mayor peso, tal y como se muestra en la tabla 2.



Estructura de las direcciones IP

Tabla 2

De aquí surge una clasificación en 5 tipos de redes en función del contenido de cada uno de los campos de dirección, tal y como se muestra en la tabla.

De esta forma, se logra una mayor optimización en las tablas de encaminamiento de los Routers y Gateways, puesto que únicamente tienen que localizar la porción de la red a la hora de encaminar un datagrama.

Dentro del direccionamiento IP, al igual que en las direcciones MAC, existe una dirección de **Broadcast** definida con todos los bits a 1 correspondientes a la porción de máquina. Es decir, la dirección 134.215.255.255 sería una dirección de Broadcast perteneciente a la red 134.215. A diferencia de MAC, dentro de IP **las redes también poseen direcciones** que se obtienen con todos los bits de la porción de máquina a 0. Continuando con el ejemplo anterior, la dirección 134.215.0.0 correspondería a la dirección IP de la red 134.215.

Cada interface IP situado dentro de un misma máquina, tiene una dirección propia IP. Significa que si en nuestro ejemplo tenemos una tarjeta de red en el servidor, y una conexión SLIP asociada a uno de sus puertos serie, éste presentará por tanto dos direcciones IP. Podríamos acceder a él a través

de cualquiera de ellas siempre que sus tablas de enrutamiento lo permitiesen.

4.2 SLIP

«Serial Line IP», es una forma sencilla de encapsulación de datagramas IP sobre líneas serie, especificado en la RFC 1055. La popularidad de SLIP se debe a las conexiones de sistemas domésticos sobre Internet a través del puerto serie RS232 existente en todos los ordenadores personales. Existe una variante denominada CSLIP o SLIP comprimido que reduce las cabeceras tanto de TCP como de IP de 40 a 3 o 5 bytes, especificado en la RFC 1144 (Van Jacobson).

El formato de la trama SLIP es sencillo. Consiste únicamente en dotar al datagrama IP de unos caracteres de comienzo y final C0, y de un sistema de transparencia que se obtiene mediante el carácter de Escape Slip DB.

Si dentro del datagrama IP aparece un carácter C0, éste se sustituye por la pareja de octetos DB,DC.

Si aparece un carácter de Escape DB, se sustituirá por dos octetos correspondientes a DB,DD.

SLIP no proporciona ningún algoritmo de control de errores similar al campo CRC de la trama Ethernet. Si una línea tele-

fónica con una relación señal ruido muy baja corrompe un datagrama transferido por SLIP, deben ser las capas superiores las encargadas de detectarlo.

Este problema queda paliado en parte, con la aparición de las normas V42 y MNP4 que permiten la corrección de errores entre ambos módem de forma transparente.

4.3 PPP

«Point to Point Protocol», es una variante más moderna de SLIP que introduce las siguientes mejoras:

* Soporte tanto para líneas asíncronas de 8 bits como de enlaces síncronos orientados a bit.

* Protocolo de control de enlace (LCP). Para establecer, configurar y comprobar la conexión del enlace de datos.

* Protocolos de control de red (NCP). Por ejemplo, NCP IP permite a cada extremo especificar si se autoriza la compresión de la cabecera igual que CSLIP.

Tanto Slip como PPP, puesto que son **conexiones punto a punto,** *deben ser considerados como redes independientes* compuestas únicamente por dos interfaces IP uno a cada lado de la conexión.

4.4 Interface de bucle local (Loopback Driver)

Este interface permite a un cliente y un servidor situados sobre la misma máquina, comunicarse entre sí vía TCP/IP. Para ello se reserva el identificador de red de clase A 127. Por convención, la mayor parte de sistemas asignan la dirección IP 127.0.0.1 al «Loopback Driver» utilizando el nombre «localhost».

Un datagrama IP enviado al interface de bucle local no aparecerá en ningún segmento de la red, produciendo una entrada IP en la misma máquina.

4.5 Máscara de Subred

Todo interface IP, necesita como mínimo dos parámetros: La dirección IP y su máscara asociada.

La máscara, se compone de 32 bits. Estos se superponen bit a bit a la dirección IP de tal forma que aquellos cuyo valor es 1, indican que la porción correspondiente a la dirección, es la parte de red. El valor 0, señala la parte de máquina. Lógicamente, existe siempre una máscara por defecto asociada a la dirección IP, en función de la clase.

En la figura 7.1, la máscara por defecto representada en notación decimal sería 255.255.0.0.(Clase B). Ello quiere

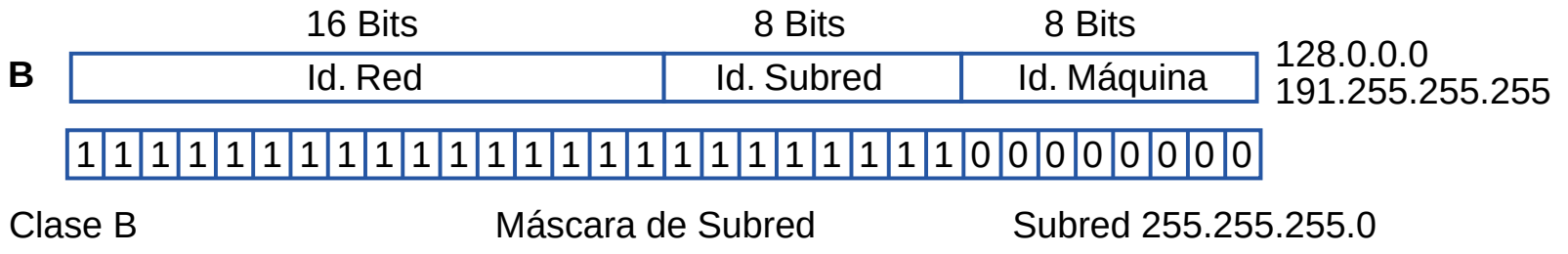


Figura 7.1

decir que nunca vamos a poder utilizar menos de 16 bits para el campo de red, aunque siempre tendremos la posibilidad de aumentar los bits de la máscara por defecto para crear lo que llamamos **subredes**. Todas las máquinas actuales soportan el direccionamiento de subred (RFC 950).

Por ejemplo, la dirección 10.2.45.1 pertenece a la red 10.0.0.0 de clase A (Ver tabla 2). Su máscara por defecto deberá ser 255.0.0.0 en notación decimal o

11111111.00000000.00000000.00000000 en notación binaria.

En un único segmento ethernet, resulta muy sencillo. Todas las máquinas conectadas llevarían la máscara 255.0.0.0 y se numerarían 10.2.45.1, 10.7.23.124, 10.0.12.253, etc., manteniendo la porción de la red siempre igual a 10. Se dispondría por tanto de 2^{24} máquinas menos 2: La dirección de broadcast 10.255.255.255 y la dirección de la red 10.0.0.0 no válidas para numerar máquinas.

Pero si quisiéramos conectar nuestro segmento con dos segmentos más, a través de una pasarela (Router), necesitaríamos ampliar la máscara como mínimo 2 bits más para tener así 4 subredes. De este modo quedaría una máscara de

11111111.11000000.00000000.00000000 o 255.192.0.0.
Dispondríamos en este caso de las siguientes subredes:

00001010.**00**000000.00000000.00000000 ó 10.0.0.0

00001010.**01**000000.00000000.00000000 ó 10.64.0.0

00001010.**10**000000.00000000.00000000 ó 10.128.0.0

00001010.**11**000000.00000000.00000000 ó 10.192.0.0

El número de máquinas por cada una de estas subredes sería 2^{22} menos 2. Por tanto, cada vez que se amplía la máscara, se pierden 2 direcciones IP en cada subred (Broadcast y red).

Resumiendo, hemos considerado que una dirección IP está compuesta de dos identificadores, uno para la red y otro para la máquina. El ámbito de cada uno de ellos depende de la clase a la que pertenece esa dirección. En realidad, el identificador de máquina puede igualmente reagrupar un identificador de subred.

Ello tiene sentido puesto que las direcciones de clase A y B reservan demasiados bits al identificador de máquina. Nadie necesita conectar tantas máquinas a una única red.

Después de haber obtenido el identificador de red de una determinada clase, es responsabilidad del administrador local del sistema decidir si es necesario o no crear una subred. En caso afirmativo, también deberá estimarse el número de bits para identificar a la subred y a la máquina. En el ejemplo siguiente se parte de una dirección de clase B, la 140.252, y sobre los 16 bits restantes, 8 son atribuidos al identificador de subred y 8 al identificador de máquina.

La utilización de 8 bits como máscara de subred suele ser comúnmente utilizada por su facilidad de comprensión, pero ello no impide que se trabaje con otros formatos. Por ejemplo, la máscara 255.255.255.224 define una subred de 11 bits y 5 bits para la dirección de máquina.

4.5.1 Máscara de longitud variable

La decisión de ampliar la máscara nos permite crear subredes y asignar así números de subred a máquinas que comparten el mismo nivel de enlace. Es lo que hemos estado viendo hasta el momento.

Pero, puede darse el caso de que alguna de las subredes creadas y a la que se le ha asigando una dirección de red, necesite dar servicio a un nuevo segmento que desea unirse a todo el conjunto.

El administrador de esta nueva subred ya dispone de una máscara asignada previamente por el sistema. No le queda otro remedio que ampliar la máscara a 13 bit por ejemplo para así asignar un número de subred al nuevo segmento.

En su máquina existirían dos interfaces con máscaras distintas: una de 11 bit que mantenía hasta el momento, y otra de 13 que estaría conectado al nuevo segmento.

El número de bits utilizados para la ampliación de la máscara dependerá de lo que estimemos oportuno en función del número de subredes a añadir y del número de máquinas.

En el ejemplo de la figura 7.2 queda reflejada esta situación. En este caso la dirección de red impuesta por el administrador que da el acceso a Internet es 140.252.0.0. Es una clase B y por tanto disponemos de 16 bit para direccionar máquinas. Se opta por una máscara de 8 bit para asignar subredes. En el esquema sólo se representan dos: la 140.252.1.0 y la 140.252.13.0. Ocurre que el administrador del segmento donde se encuentran las máquinas A y C necesita enlazar la

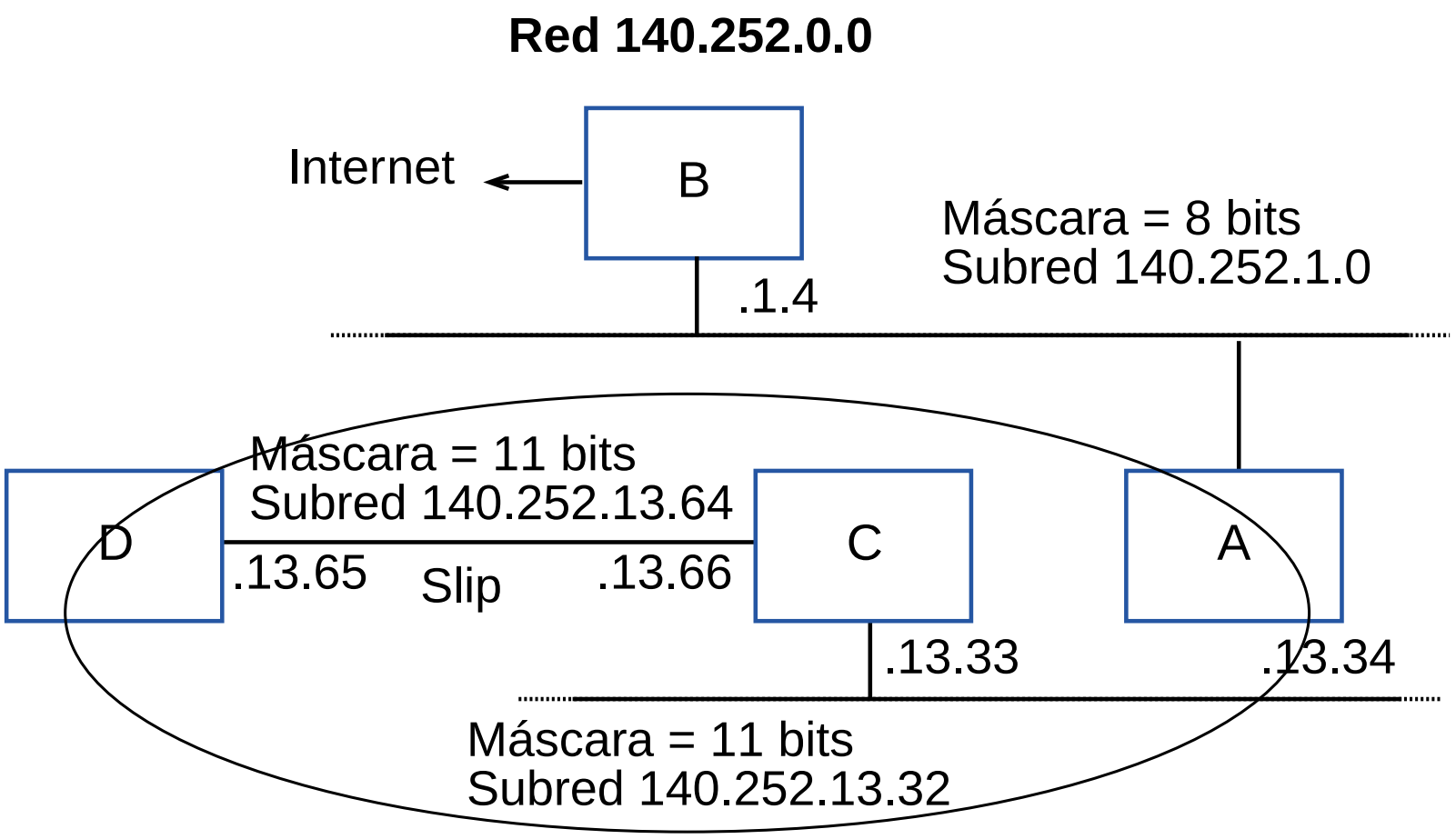


Figura 7.2

máquina D mediante una conexión Slip. Su solución es ampliar la máscara dentro de su dominio a 11 bit para poder así disponer de subredes.

Cabe resaltar que la máquina B y el resto del sistema sólo ven una subred en ese dominio, la 140.252.13.0. Las máscaras de la máquina A son diferentes, una es de 8 bit y la otra de 11. Puede comprobarse que no existirá ninguna dirección IP repetida.

4.6 Cabecera IP

La figura 7.3 muestra el formato del datagrama IP. El tamaño normal de una cabecera IP es de 20 bytes, a no ser que presente opciones.

El bit más significativo está marcado como 0 en el lado izquierdo, mientras que el menos significativo de la palabra de 32 bits se etiqueta como 31 en el lado derecho. Los 4 octetos de cada palabra de 32 bit se transmiten empezando por el 0 hasta el 31.

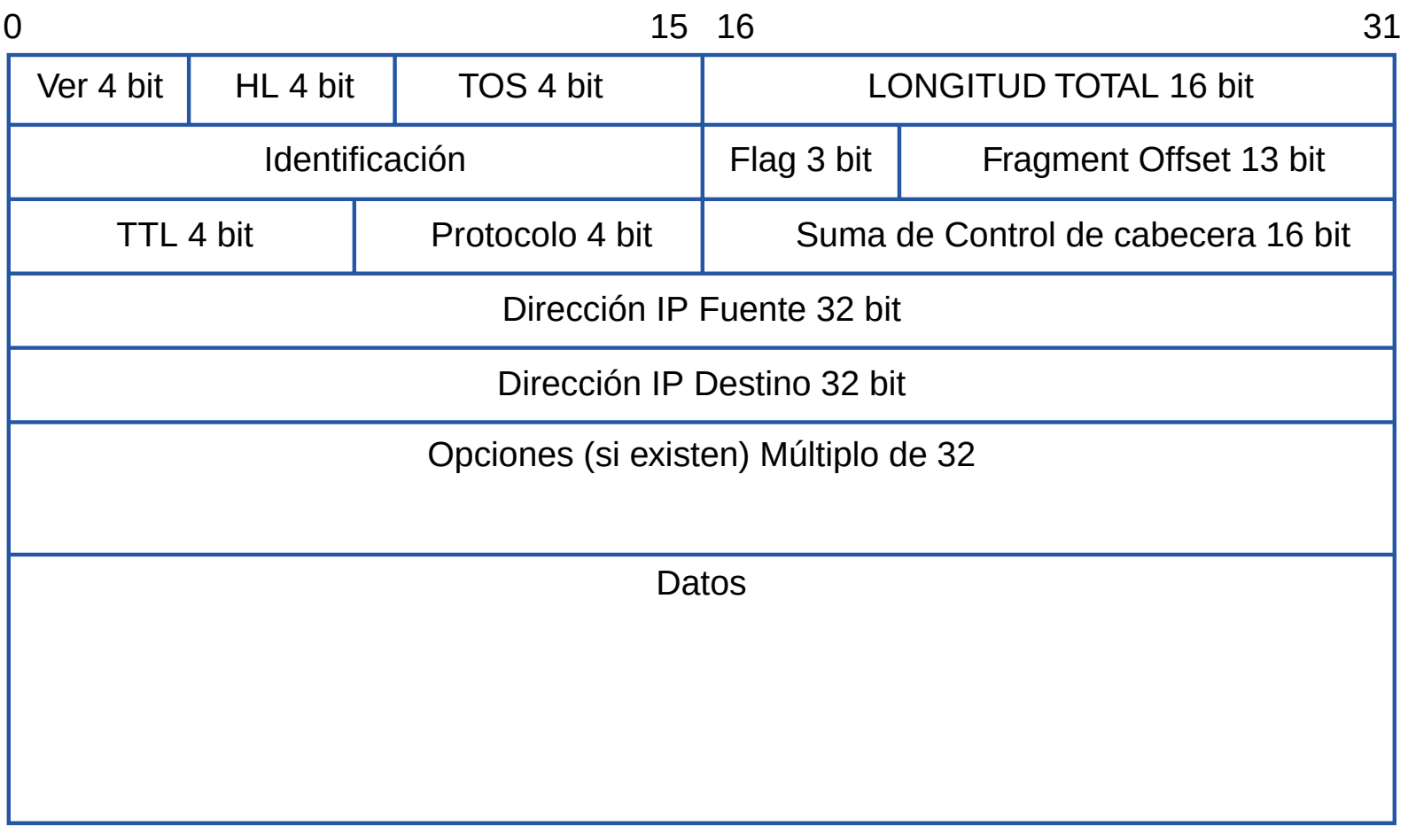


Figura 7.3

La versión en curso actualmente es la 4 también conocida como **IPv4**. El campo «*Ver*» sobre 4 bit transporta esta información. Existen actualmente en desarrollo cuatro propuestas para la aparición de una nueva versión de IP: SIP (Simple Internet Protocol RFC 1347), PIP, TUBA (TCP and UDP with Bigger Address) y TP/IX (RFC 1475). Como únicamente podrá sobrevivir una de ellas, aquella que proporcione la mejor alternativa a IPv4 se denominará IPv5 o IPng (IP nueva generación).

El campo HL indica el número de palabras de 32 bit que componen la cabecera, incluyendo las opciones eventuales. Puesto que su tamaño es de 4 bit, tendremos que 2^4 x 32 bit por palabra, son 64 bytes de longitud máxima en la cabecera IP. Este campo posee habitualmente el valor 5 (Cuando no existen opciones).

TOS (Type of service) indica el Tipo de Servicio. Actualmente los 3 primeros bits son ignorados, los 4 siguientes representan el TOS y el último está inutilizado y su valor debe ser siempre 0.

Los cuatro bits que componen este campo quedan descritos en la figura 7.4. Únicamente uno de ellos puede estar posicionado a 1. Si todos están a 0, esto significa un servicio nor-

Aplicación	Minimiza Retardo	Maximiza Caudal	Maximiza Fiablilidad	Minimiza Coste
Telnet/Rlogin	1	0	0	0
FTP				
Control	1	0	0	0
Datos	0	1	0	0
TFTP	1	0	0	0
SMTP				
Comando	1	0	0	0
Datos	0	1	0	0
DNS				
Petición UDP	1	0	0	0
Petición TCP	0	0	0	0
Transf. Zona	0	1	0	0
ICMP				
Error	0	0	0	0
Petición	0	0	0	0
SNMP	0	0	1	0
BOOTP	0	0	0	0
NNTP	0	0	0	1

Figura 7.4

mal. La RFC 1340 especifica cómo deben ser activados por el conjunto de las aplicaciones estándar. La RFC 1349 aporta algunas correcciones y una descripción más detallada de las particularidades del TOS.

La figura 7.4 indica los valores recomendados para diversas aplicaciones. Aquellas orientadas a una conexión interactiva como Telnet o Rlogin, necesitan tiempos de respuesta mínimos puesto que proporcionan un diálogo entre un ser huma-

no y una máquina. Las transferencias de ficheros como FTP, requieren un caudal máximo para el enlace de datos.

La particularidad de TOS no está soportada por la mayoría de implementaciones TCP/IP actuales, únicamente sistemas modernos con versiones posteriores a la 4.3BSD lo utilizan. Los protocolos de encaminamiento más recientes como OSPF e IS-IS son capaces de tomar decisiones en función del valor de este campo.

El campo **Longitud Total** contiene el tamaño en octetos del datagrama IP. Gracias a él y al campo HL podemos conocer donde empieza y termina la porción de datos. Como utiliza 16 bit, se puede deducir que el tamaño máximo o MTU de un datagrama IP será de 65535 bytes.

El mecanismo de fragmentación utilizado por IP hace uso de los siguientes 3 campos. El primero, **Identificación,** permite marcar de forma única cada datagrama enviado por una máquina. Se incrementa normalmente en cada nuevo envío. Cuando se produce una **fragmentación**, este valor es copia-do en cada uno de los trozos o fragmentos que componen el datagrama original. El campo **flag** de 3 bit, activa entonces uno de ellos conocido como *«more fragments»* colocándolo a 1 en todos los trozos excepto en el último. El campo

«Fragment Offset» contiene el índice del fragmento a partir del datagrama original. Además, el campo Longitud Total de cada fragmento es actualizado a su nuevo valor.

Existe un bit en el campo *«Flag»* conocido como *«don't fragment»*. Si está activado a 1, IP no producirá ninguna fragmentación eliminando el datagrama y enviando un mensaje de error ICMP a la fuente.

Para evitar que un datagrama quede atrapado en algún bucle dentro de la red (Problemas con los protocolos de encaminamiento, p.ej.) existe un tiempo de vida representado mediante el campo **TTL** (Time to Live). Se inicializa a un cierto valor por el remitente y se decrementa en una unidad por cada Pasarela o Router que atraviesa. Cuando es alcanzado el valor 0, el datagrama se elimina y un mensaje ICMP es enviado a la fuente indicando el suceso.

IP identifica el protocolo (TCP, UDP, ICMP…) al cual debe hacer llegar la información, a través de campo **Protocolo**.

La **Suma de Control** abarca únicamente la cabecera IP. Se calcula como una suma sin acarreo sobre 16 bit, de todos los bytes que componen la cabecera IP considerándolos como una secuencia de palabras de 16 bit. Sin embargo, otros pro-

tocolos como TCP, UDP, ICMP utilizan códigos de redundancia cíclica (CRC) basados en algoritmos más sofisticados.

El motivo es claro. Una Pasarela o Router debe procesar grandes cantidades de paquetes por unidad de tiempo. Generalmente, el único valor que modifica a cada datagrama es el TTL, decrementándolo en una unidad. El cálculo de la suma de control puede ser realizado de forma incremental disminuyendo drásticamente el tiempo de proceso de cada datagrama por las pasarelas intermedias.

Como ya se comentó anteriormente, cada datagrama contiene la **dirección IP** del destinatario y la del remitente.

El campo **Opciones** es una lista de longitud variable con información específica del datagrama. Las opciones actualmente definidas son las siguientes:

- Seguridad y gestión de restricciones. Para aplicaciones militares documentado en la RFC 1108.
- Registro de ruta. Cada pasarela puede añadir su IP.
- Estampilla horaria. Cada pasarela puede añadir su IP y el tiempo.
- Enrutamiento no estricto de la fuente. Lista de direcciones IP por las que debe atravesar el datagrama.

- Enrutamiento estricto de la fuente. Similar al caso anterior pero la lista de direcciones es obligatoria.

Estas opciones son raramente utilizadas y no todas las máquinas las soportan. Las direcciones del campo **Opciones** acaban siempre con fronteras de 32 bits. En caso necesario pueden ser añadidos bits de relleno con el valor 0.

4.7 Administración de una red IP

A la hora de asignar direcciones a las máquinas de una red con protocolo IP, podemos encontrarnos básicamente con dos situaciones.

1. Nuestra red no está conectada con una Internet y por tanto no depende de ningún direccionamiento existente.
2. Tenemos un acceso a una Internet a través de un Router y el administrador del sistema nos ha asignado una dirección de red (A, B o C) para que la administremos a nivel local.

Tanto en un caso como en el otro tendremos que tener claro cuántas subredes necesitamos.

Las conexiones Punto a Punto son un caso especial, ya que dada su naturaleza de medio no «Broadcast» pueden ser agrupadas dentro de una misma subred en el caso de que

nuestro sistema posea más de una conexión punto a punto. Las tablas de encaminamiento cuando se trata de enlaces punto a punto siempre hacen referencia a la ruta especificando la dirección completa de cada extremo de la conexión.

En el primer caso, la solución es sencilla, podemos utilizar una Clase A, B o C con la máscara por defecto. Simplemente tendremos que evitar asignar dos subredes físicamente diferentes con el mismo identificador de red. Cada subred debe tener un número de red diferente.

En el segundo caso, dependemos de una dirección impuesta por el administrador de la red a la que nos hemos de conectar. No podemos alterar el campo de red. Sin embargo, podemos ampliar la máscara por defecto asociada a la clase para crear las subredes que necesitemos, tal y como se ha explicado en las secciones precedentes.

5. Caso práctico

En la figura siguiente queda representada con más detalle la configuración de referencia sobre la que se realizarán los ensayos, así como sus respectivas direcciones IP.

En nuestro ejemplo, el servidor **«std»** tiene en su interface Ethernet la dirección **128.128.128.254** y la máscara con la que se ha configurado el sistema es 255.255.255.0. De aquí puede deducirse que se trata de una red de clase B, donde existe una subred con un campo de 8 bits y una porción de máquina de 8 bits.

Aunque debido a la máscara de subred utilizada podríamos utilizar hasta 128 subredes, en nuestro ejemplo sólo tenemos tres.

Dos de ellas pertenecen a segmentos Ethernet aislados entre sí, son las numeradas como .128.0 y .127.0, aunque una de

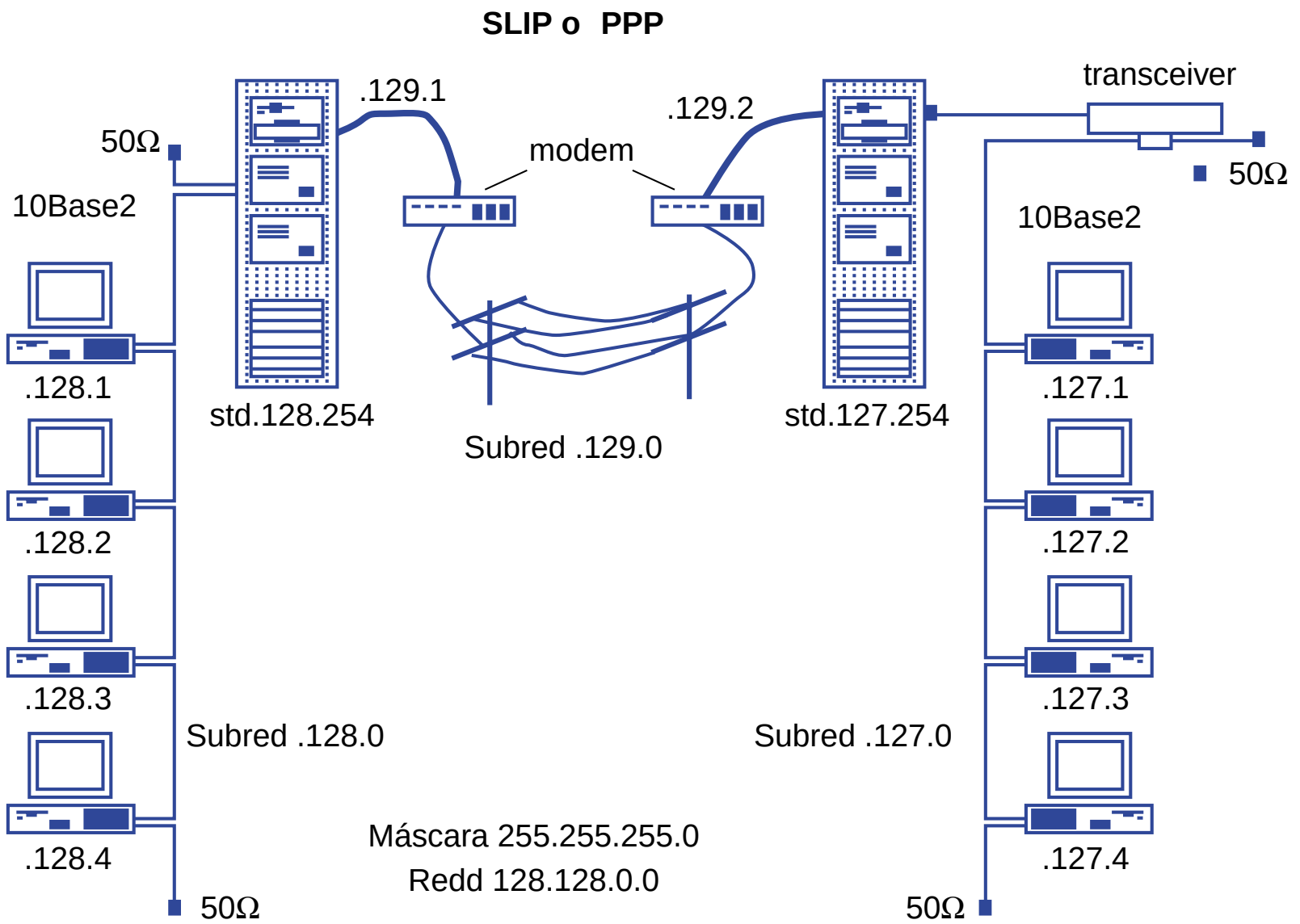


Figura 8

ellas pudiera haber sido un anillo Token Ring por ejemplo. La tercera subred corresponde a la conexión Slip con la numeración .129.0, puesto que este tipo de enlaces requiere una dirección IP a cada extremo.

La notación .128.4 o .129.0 indica que la dirección ha de ser completada (al empezar por un punto) y su objetivo es sim-

plificar la representación gráfica. Se sobreentiende que las direcciones reales serían 128.128.128.4 y 128.128.129.0.

Una red o subred está formada por un conjunto de máquinas que comparten el mismo medio de transmisión. Este puede ser de tipo multipunto o *«broadcast»* como lo es Ethernet, o sencillamente estar constituida por un enlace punto a punto en el que intervienen únicamente dos estaciones, tal es el caso de la conexión Slip.

Los Routers se caracterizan por disponer de dos o más interfaces de red, además del software de encaminamiento necesario. Cada interface debe disponer de una dirección de red distinta a los demás. El servidor **std2** tiene dos direcciones IP correspondientes a redes diferentes, al igual que **std**.

Si deseáramos unir nuestro sistema a un anillo Token Ring, bastaría con añadir una tarjeta Token Ring al servidor correspondiente y asignarle la subred .124.0 por ejemplo.

Si una estación desea enviar un datagrama IP a otra situada en su misma red, lo hará enviando una trama ethernet con la dirección MAC correspondiente a la estación destino. Si el datagrama va dirigido a otra red diferente, se confecciona una trama ethernet con la dirección MAC del Router por defecto, identificado a menudo como *«Default Gateway»*.

En el primer caso, la trama ethernet contiene la dirección MAC del destinatario, y el datagrama que transporta también posee la dirección IP del mismo.

En el segundo caso, la trama ethernet enviada contiene la dirección MAC del Router por defecto, pero la dirección IP del datagrama transportado es la del destinatario.

Puesto que los dos servidores Unix están configurados como Routers o Gateways, esta circunstancia deberá ser considerada al configurar el parámetro *«Default Gateway»* en las estaciones de trabajo.

Es necesario tener presente que a nivel de direccionamiento IP, el tipo de protocolo de enlace utilizado es independiente. Es decir, la conexión vía módem representada en la figura 1, puede ser sustituida por una conexión a través de una red pública de datos como Iberpac con el protocolo de enlace X25, un enlace a través de RDSI, Frame Relay etc. Todo ello sin alterar para nada el direccionamiento IP establecido.

6. Configuración de TUN (MS-DOS)

En primer lugar, debe comprobarse que nuestra estación se encuentra conectada correctamente al segmento de la red. Seguidamente nos hemos de posicionar en el directorio C:\TUNTCP y lanzar un ejecutable denominado TUNTCP.EXE. Se presentará un menú en la pantalla con diversas opciones de las que seleccionaremos, de momento, la primera o TCP/IP. EL primer paso es crear la tabla de servidores con las direcciones IP correspondientes. Esta tabla se sitúa en el directorio \TUNCTP y se llama HOSTTAB. No es una operación imprescindible, pero facilita el uso de posteriores comandos puesto que resulta más fácil referenciar una estación o un servidor por un nombre que por su dirección IP.

El paso siguiente será la configuración del núcleo de TCP/IP. En «Parámetros de Lanzamiento» se accede a las opciones

de configuración. Todos los parámetros solicitados han sido expuestos con anterioridad, a excepción del tipo de tarjeta, donde deberá referenciarse el modelo Etherlink I de 3Com o 3C501, admitiendo todas las opciones por defecto.

Se deberán confirmar los cambios pulsando la tecla de función F2, lo cual provocará la aparición de un mensaje en pantalla indicando modificaciones en el fichero CONFIG.SYS. Una vez realizado esto, es necesario salir del programa mediante la tecla Escape repetidas veces y arrancar de nuevo la máquina.

De nuevo nos posicionamos sobre C:\TUNTCP y ejecutamos ahora el fichero AUTOTCP.BAT. Aparecerán una serie de mensajes entre los que figurará la dirección MAC de nuestra tarjeta, que anotaremos para posteriores consultas. Si esto no ocurre así, es muy probable que tengamos mal configurada la tarjeta.

En este punto se puede comprobar que la mayoría de las tarjetas instaladas en la sala pertenecen al mismo fabricante (3Com).

Una forma rápida de conocer si estamos en red, es mediante el uso del comando PING.EXE desde el «prompt» de MSDOS. Existe una versión más sofisticada en Unix que luego veremos. El modo de utilizarlo es el siguiente:

PING <nombre de la estación dada de alta en HOSTTAB> o

PING <dirección IP>

Si todo está correcto recibiremos el mensaje «host responding». Es importante comenzar por estaciones que se encuentren sobre el mismo segmento físico de red, para luego intentarlo sobre otras estaciones situadas al otro lado de la conexión **SLIP**.

Posteriormente profundizaremos un poco más sobre el programa PING.

Puesto que, suponiendo que todo haya ido bien, nuestro enlace IP funciona correctamente, vamos a comprobar que la capa TCP está operativa. Es necesario tener presente que «Ping» no utiliza protocolos de transporte y se basa únicamente en IP e ICMP, con lo que aunque funcione correctamente, no existe garantía de que los protocolos por encima de IP también lo hagan (Ver figura 9).

Ejecutamos para ello desde la máquina local en MS-DOS:

C:\TUNTCP> REXEC STD -L login -P password <comando Unix>

Ello retorna la salida del comando Unix invocado mediante el servicio «rexec», hacia la pantalla en modo MS-DOS.

7. Cuestionario

1º) ¿Qué ventaja puede aportarnos el disponer de un interface AUI en nuestra placa de red?

2º) ¿Qué ocurre si se intenta enviar un bloque de información de tamaño superior al MTU de la capa de enlace?

3º) ¿A cuántos tipos de direcciones destino es capaz de responder una tarjeta de red?

4º) ¿Cuántas máquinas podrían ser direccionadas con los siguientes parámetros de red?

Dirección de la red: 93.0.0.0

Máscara de subred: 255.255.255.0

5º) Determinar cuantas subredes podrían ser generadas en la cuestión anterior.

6º) Calcular las direcciones IP y las máscaras de subred de cada uno de los interfaces de la red 203.109.44.0 representada en la figura 8.1.

7º) ¿Qué sucede si se cambia la máscara de subred al valor por defecto (0 bits de subred) en una estación conectada al segmento Ethernet?

Razónense los hechos observados.

8º) ¿Qué pasa si el Router por defecto no es el adecuado o no existe?

Red 203.109.44.0
masc: 255.255.255.0

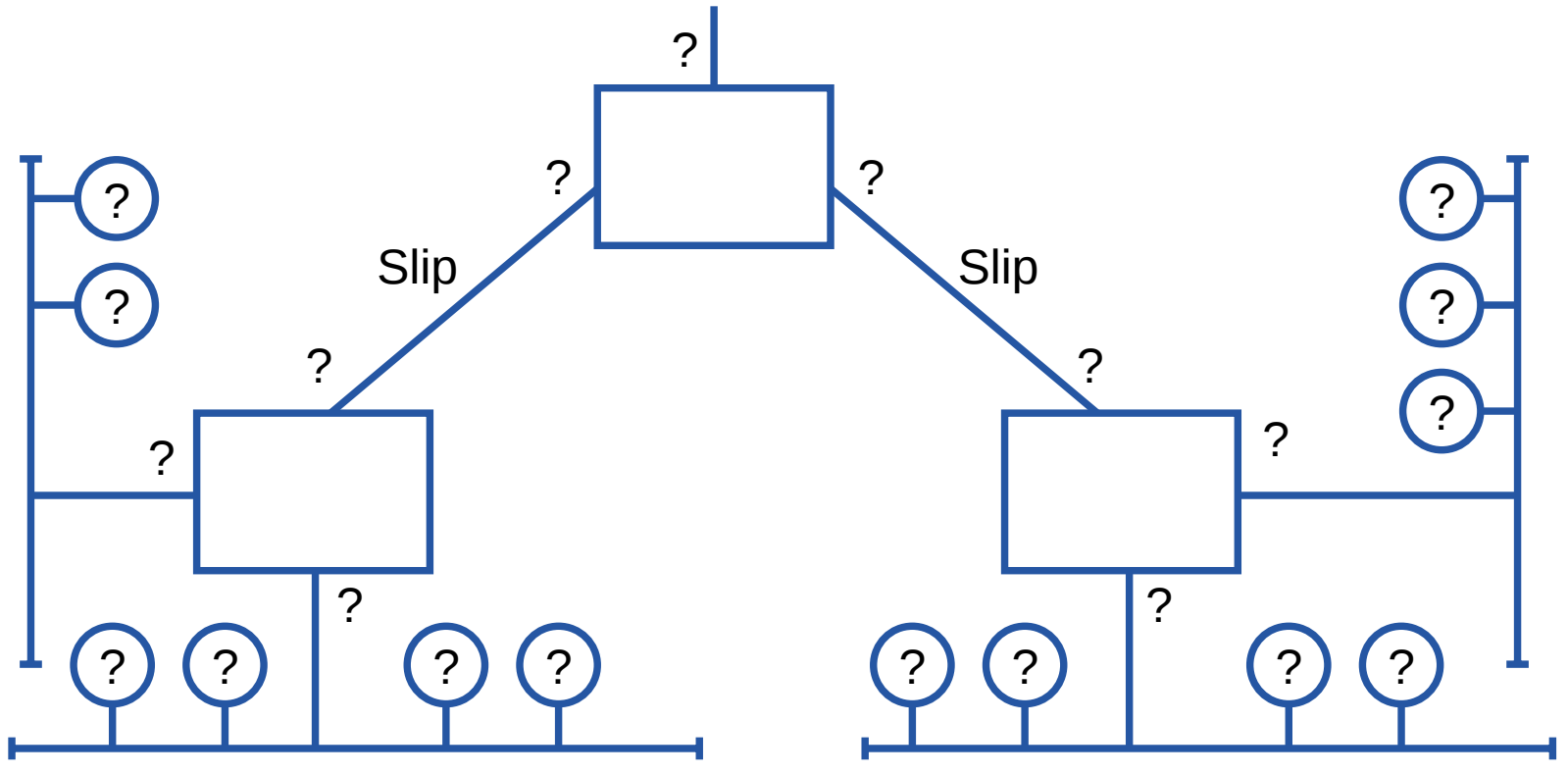


Figura 8.1

8. Estructura de la serie de protocolos TCP/IP

En la figura siguiente se pueden observar las distintas capas que forman la serie de protocolos TCP/IP, así como sus interacciones:

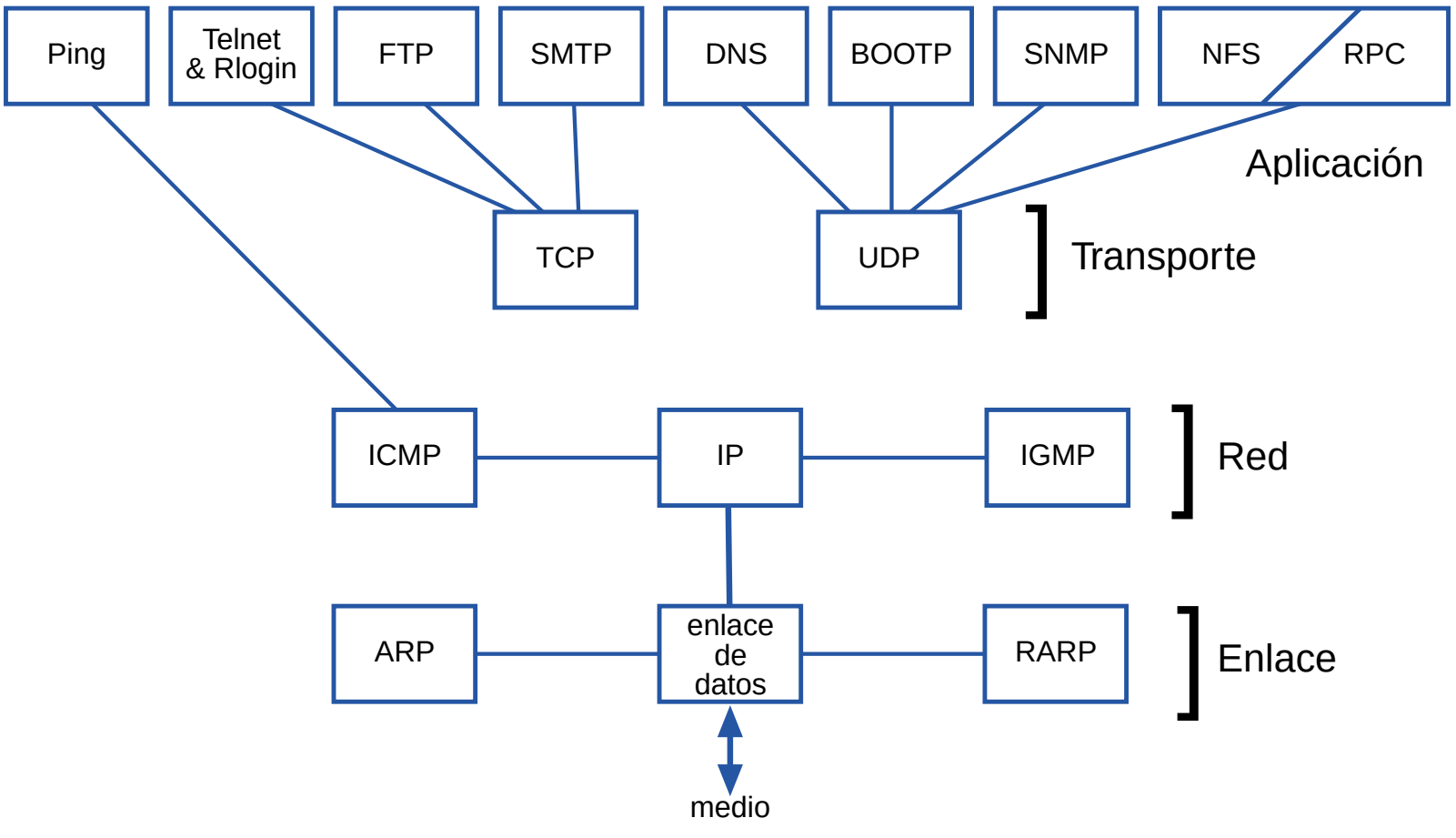


Figura 9

Los protocolos de red están generalmente organizados en capas, de forma que cada una de ellas controla una parte de las comunicaciones. **TCP/IP** es una **combinación de diferentes protocolos** situados en capas diferentes.

IP es el «caballo de batalla» utilizado por todos los protocolos de la estructura. El conjunto de datos emitidos por TCP, UDP, ICMP y IGMP son transmitidos como datagramas IP. El hecho de que IP proporcione un servicio no fiable, utilizando una entrega de datagramas sin conexión, sorprende muchos allegados a TCP/IP, especialmente aquellos que han conocido X.25 o SNA.

Por «no fiable», se entiende que no existe ninguna garantía de que el datagrama llegue a su destino. IP proporciona un servicio con el mínimo consumo de recursos. Cuando ocurre algún problema, como por ejemplo una saturación en los «buffers» de cualquier Router intermedio, IP utiliza un algoritmo de gestión muy simple: devuelve el datagrama intentando enviar un mensaje de control ICMP a la fuente.

Cualquier exigencia de fiabilidad debe ser asegurada por los niveles superiores, por ejemplo TCP.

Observando el diagrama podemos comprobar que hasta el momento no hemos echo en absoluto uso de los protocolos

de transporte **TCP** o **UDP**. Únicamente han entrado en funcionamiento los protocolos **ICMP** y **ARP** que describiremos en breve.

8.1 Número de Puerto

TCP y UDP identifican las aplicaciones utilizando números de 16 bits. Existen una serie de puertos claramente especificados para determinados servicios. Por ejemplo, cada implementación TCP/IP que genera un servidor FTP, lo hace a través del puerto TCP 21, cada servidor Telnet figura en el puerto TCP 23, y cada implementación de TFTP reside sobre el puerto UDP 69.

Aquellos servicios proporcionados por implementaciones TCP/IP, tienen números de puerto **bien definidos** comprendidos entre 1 y 1023, y son gestionados por la IANA (Internet Assigned Numbers Authority). Los números de puerto de los clientes son **efímeros** puesto que éstos únicamente existen mientras necesitan hacer uso del servicio. Se reservan por tanto, en la mayoría de versiones TCP/IP los puertos **efímeros** del 1024 al 5000. Por encima del puerto 5000 se reservan a otros servidores no definidos sobre Internet.

8.2 Socket

Cada segmento TCP contiene los números de puerto fuente y destino con el fin de identificar la aplicación emisora y la receptora. Estos dos valores, combinados con las direcciones IP fuente y destino, identifican cada conexión de forma única. La combinación de un número de puerto y una dirección IP se conoce como **socket.**

Por tanto, el cuarteto compuesto por la dirección IP del Cliente, el número de puerto del Cliente, la dirección IP del Servidor y el número de puerto del Servidor; forma la pareja de sockets que identifican cada conexión TCP.

8.3 Pequeño Inciso

Antes de continuar, vamos a dar un pequeño salto en el orden lógico de exposición de la materia, ejecutando un servicio que nos permitirá tener acceso directo sobre los servidores Unix. Se trata del servicio **Telnet**, implementado por TUN mediante el ejecutable TNVT52.EXE que proporciona una sencilla emulación tipo VT52 suficiente para nuestras pretensiones.

TUN posee un producto adicional no instalado en nuestro caso, pero utilizado de forma asidua por todos los usuarios de

TUNTCP denominado TUNEMUL, que permite cuatro sesiones simultáneas con emulación tipo ANSI.

La forma de ejecutar TNVT52.EXE es muy similar a la de PING.EXE, es decir:

TNVT52 <nombre de servidor> o

TNVT52 <dirección IP> o

TNVT52

En el último caso, el programa preguntará el nombre o la dirección IP del servidor al cual queremos conectarnos. El esquema de funcionamiento de Telnet queda reflejado en la figura 10.

Lógicamente, para poder acceder de cualquier forma a un sistema Multiusuario como Unix se necesita tener creadas previamente las cuentas de usuario, labor que corresponde al administrador del sistema.

Los terminales de red en Unix se asocian a terminales virtuales y generalmente vienen representados como /dev/ttypn, donde n es el número de terminal asignado. Dicha asignación es dinámica, de forma que no existe ninguna seguridad de tener el mismo número de terminal en dos conexiones con-

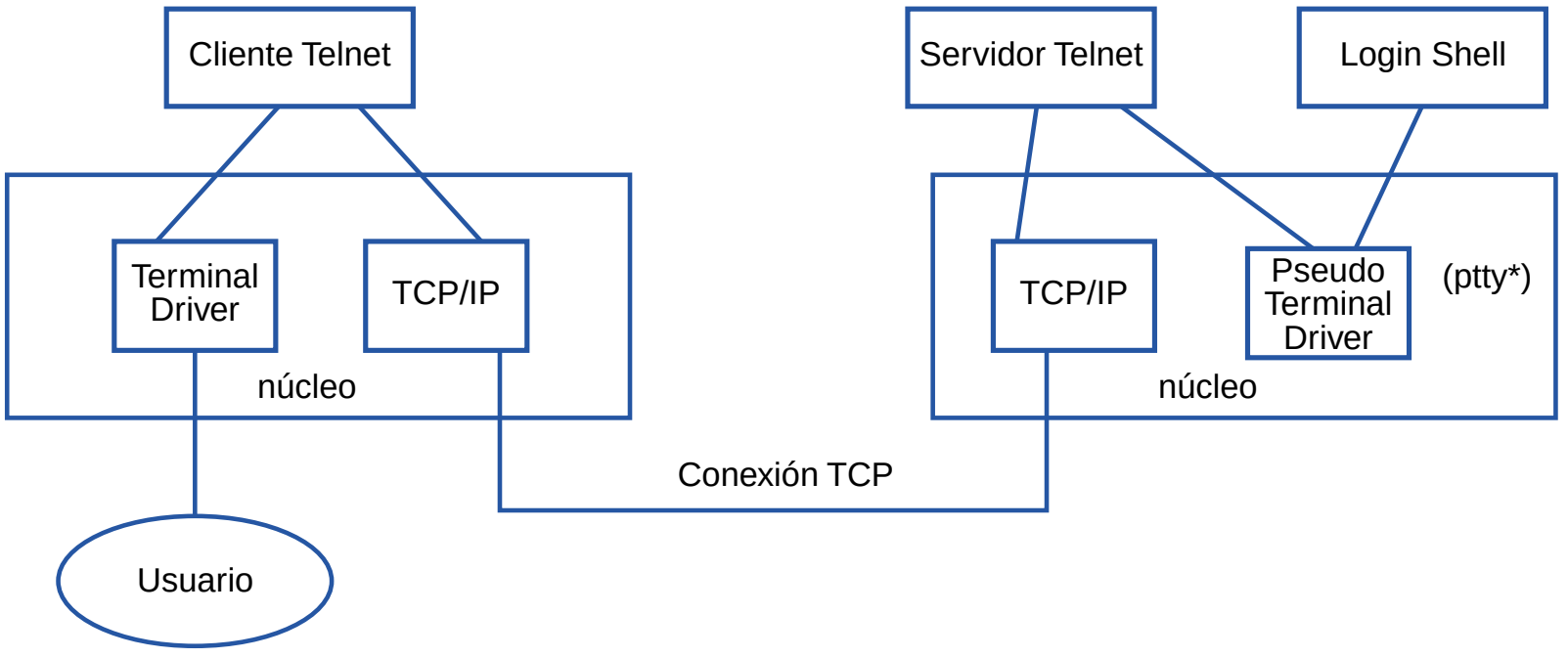


Figura 10

secutivas. Ello puede ser comprobado efectuando 2 conexiones al mismo servidor y ejecutando el comando:

$ tty

La respuesta podría ser:

$ /dev/ttyp12

Con:

$ who -u

Podremos visualizar el estado de todos los usuarios conectados y sus ttyp asociados.

Una vez conectados vía Telnet (TNVT52.EXE) a nuestro servidor local, podemos hacer de nuevo uso del programa cliente Telnet de éste último.

$ hostname permite conocer el nombre de nuestro servidor

std

$ telnet std2

login: xxxxx

Password: xxxxx

Sin embargo el servicio Telnet proporcionado por la mayoría de los sistemas Unix, permite especificar en la línea de comandos un parámetro mas: el número de puerto. Esto podemos comprobarlo intentando una conexión al puerto número 7 correspondiente al servicio *«echo»*, cuya única finalidad es la de devolver al remitente cualquier datagrama recibido.

$ telnet std2 7

En apariencia parece que no ocurre nada, sin embargo cualquier carácter pulsado será devuelto a la pantalla indicando que nuestro interlocutor está realizando su labor correctamente: devolver todo lo que llega al puerto número 7.

Puesto que utilizamos Telnet, la conexión con el puerto 7 será a través de un socket TCP. Ello nos puede servir para seguir la pista a distintos tipos de problemas que suelen aparecer en este tipo de configuraciones.

Una fuente de errores en muchas instalaciones, es la aceptación de parámetros por defecto en la generación y configuración de TCP/IP. Ello provoca generalmente la creación de un número limitado de ttypn (16 en algunos casos). Si tenemos en cuenta que cada puesto de trabajo, puede tener más de una conexión Telnet y multiplicamos por el número de estaciones veremos que 16 en nuestro caso sería insuficiente. Lo mismo ocurre con las conexiones TCP. Todo ello deberá ser corregido mediante el comando **«netconfig».**

Aprovechando la conexión con el servidor mediante Telnet, resulta interesante comprobar la existencia del proceso que controla la conexión SLIP. Este fue invocado en el proceso de arranque de la máquina por una de las «shell» de conexión situada en el directorio /etc/rc2.d denominada «S85tcp». Mediante el comando:

$ more /etc/rc2.d/S85tcp

Podemos localizar una línea que hace alusión a:

slattach +v +c tty1a 128.128.128.1 128.128.128.2 19200

(Pueden variar las direcciones IP según el servidor de donde se lance)

La opción +v indica modo «verbose» o emisión de mensajes de estado a la consola. La opción +c indica compresión de cabeceras IP y TCP tipo Van Jacobson. A continuación aparecen las direcciones IP de cada extremo de la conexión SLIP y luego se especifica la velocidad de transmisión: 19200 bps.

Mediante el comando:

$ ps -ef|grep slattach

Localizamos en la tabla de procesos la línea correspondiente a «slattach» donde figura el comando completo tal y como fue lanzado, y el número de proceso Unix.

8.4 Ping

La versión proporcionada por los sistemas Unix, OS/2, Windows 95 o NT de este programa presenta una serie de posibilidades que le convierten en una herramienta muy valiosa a la hora de depurar y localizar errores.

Se basa en el protocolo ICMP, o protocolo de control de transmisión. Ping a diferencia del resto de aplicaciones TCP/IP no utiliza ninguno de los protocolos de transporte TCP o UDP. Se apoya directamente sobre IP. Este es un hecho a tener en

cuenta, dado que la recepción de una respuesta Ping indica que la máquina remota está activa, pero no asegura que el funcionamiento de su capa TCP o UDP sea el correcto.

Ping utiliza el comando eco de ICMP para enviar un datagrama a su destinatario y esperar su retorno. De este modo es capaz de evaluar tiempos de respuesta promedios.

Dispone de varias opciones, entre las que cabe destacar la posibilidad de modificar el tamaño del paquete enviado, el registro de ruta, y el control del número de paquetes enviados.

$ ping -s 200 -c 3 std

Provocaría la emisión hacia el servidor std de 3 paquetes con 200 bytes cada uno de datos a los que habría que sumar 20 bytes de la cabecera IP y 8 de la ICMP.

La respuesta que PING proporciona en pantalla corresponde a una serie de líneas donde se indica en tiempo de respuesta del eco ICMP y el número de secuencia. Al concluir el

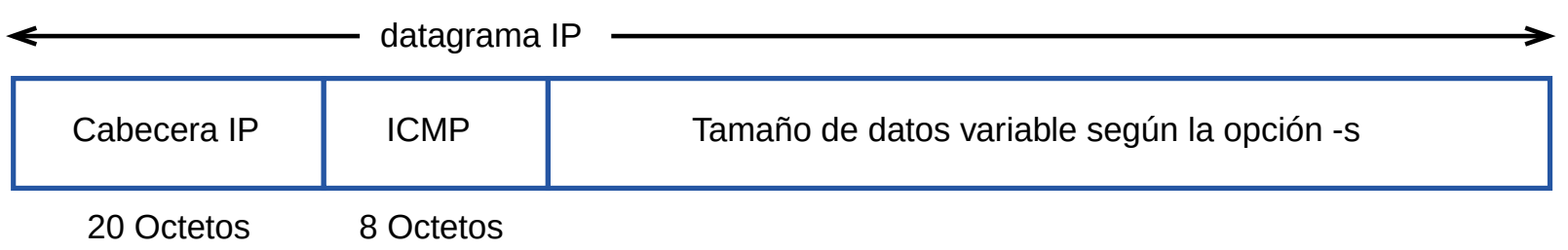


Figura 10.1

comando, queda reflejado el número de paquetes perdidos, los tiempos mínimos, máximos y medios de respuesta (ida y vuelta).

Nos permitirá conocer la tasa de error de un enlace así como la velocidad real de transmisión de forma experimental.

8.5 Servidores de terminales

Son dispositivos electrónicos que permiten la conexión de terminales asíncronos RS-232 sobre TCP/IP. En el caso de Ethernet, disponen de una dirección MAC como cualquier tarjeta de red. Además, deben ser parametrizados de forma que puedan tener su dirección IP, su máscara de subred y gateway por defecto.

Incorporan normalmente los servicios **Telnet** y **Rlogin.** Opcionalmente pueden hacer uso de **BOOTP,TFTP** y **Ping.**

Todos ellos hacen uso de la misma dirección IP del Servidor de terminales, asignándose un número de puerto TCP diferente para cada línea asíncrona. De esta forma queda perfectamente definida cada conexión.

Realmente un Servidor de Terminales no es más que un **cliente Telnet,** que proporciona sesiones de forma externa mediante líneas RS-232.

La utilización de estos dispositivos queda justificada por el hecho de que la mayoría de usuarios que pretenden migrar a entornos de Red Local, disponen con frecuencia de periferia RS-232. La posibilidad de continuar aprovechando esta inversión integrándola en la red, es un poderoso argumento de venta.

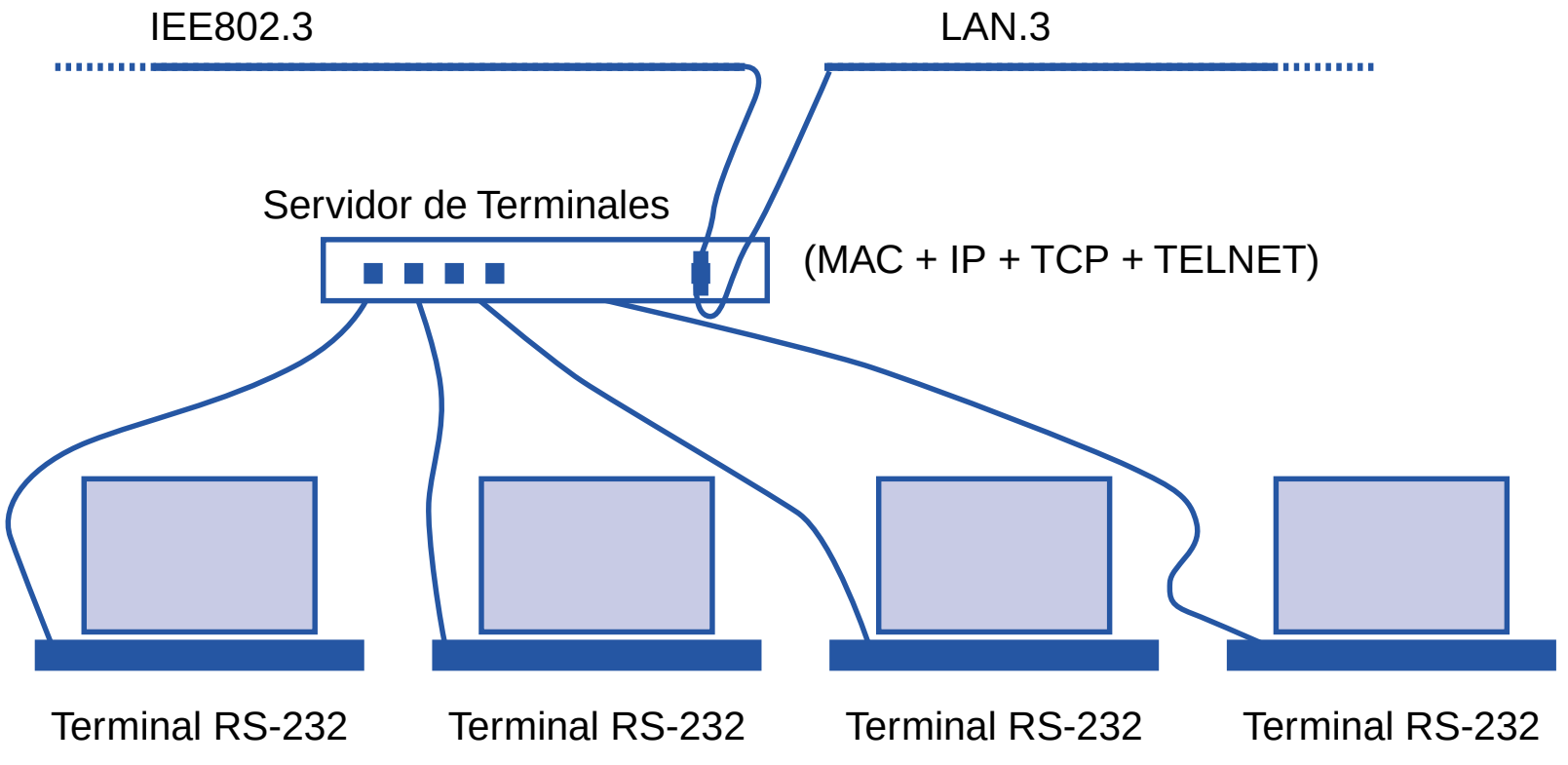


Figura 11

9. Configuración en UNIX

Vamos a utilizar los comandos Unix de red **ifconfig** y **netstat**. Para ello deberemos establecer una conexión con nuestro servidor local en un principio.

Una vez introducidos en el sistema, ejecutaremos:

$ netstat -i

La opción -i visualiza información acerca de los interfaces físicos del sistema:

Name	Mtu	Network	ddress	Ipkts	Ierrs	Opkts	Oerrs	Collis
e3A0	1500	128.128.12	std2	18	0	113	0	0
lo0	8232	loopback	localhost	1840	0	1840	0	0
sl0	296	128.128	std2_slip	332	334	0	0	0

La primera columna indica el nombre que Unix da al interface instalado. e3A0 es el nombre de la tarjeta de red Etherlink I de 3Com, lo0 es el «Loopback Driver», y sl0 es el nombre de la conexión SLIP realizada a través del dispositivo tty1a (COM1 en MSDOS). Estos han sido creados previamente en el proceso de instalación mediante el comando **«netconfig»**.

La segunda columna indica el MTU que tiene asignado cada interface, y el resto presentan información adicional que comentaremos más adelante.

Para ver la configuración de cada interface haremos uso de **ifconfig**. Su sintaxis es la siguiente:

$ ifconfig <nombre del interface>

Si quisiéramos saber las características de nuestra tarjeta de red ejecutaríamos:

$ ifconfig e3A0

Obteniendo una respuesta similar a:

e3A0: flags=823<UP,BROADCAST,NOTRAILERS,ONE-PACKET>

inet 128.128.127.254 netmask ffffff00 broadcast 128.128.127.255

one-packet mode params: packet size: 512; threshold: 3

Puede observarse que el interface se encuentra en servicio (UP), que admite dirección de BROADCAST y que el tipo de trama utilizado es Ethernet sin modo Trailer.

La dirección IP correspondiente es 128.128.127.254 de clase B y existe una máscara que define una subred de 8 bits con una porción de máquina de 8 bits. Además de la dirección de broadcast, aparece el tamaño de trama y umbral de disparo para el modo One-Packet que actualmente está en desuso.

Si lo que deseamos es sondear el interface sl0 correspondiente a la conexión SLIP:

$ ifconfig sl0

El resultado será algo así:

sl0: flags=11<UP,POINTOPOINT>

inet 128.128.129.2 --> 128.128.129.1 netmask ffffff00

Como ejercicio, sería interesante ejecutar ifconfig en los dos servidores (std y std2) mediante respectivas conexiones Telnet.

El comando ifconfig, no sólo permite observar las configuraciones sino también cambiarlas. Para ello se necesitan privilegios de superusuario (root). Por ejemplo, si se quisiera cambiar de forma dinámica la máscara de subred de la tarjeta:

$ ifconfig e3A0 netmask 255.255.224.0

Se estaría definiendo una subred de 3 bits y 13 bits de máquina (.11100000.00000000). Pueden ser modificados otros parámetros además de la máscara y la dirección IP. Tal es el caso de la «métrica», factor que hace prevalecer una ruta sobre otra a la hora de encaminar los paquetes.

9.1 Tabla de hosts

Mediante TNVT52.EXE, estableceremos una conexión con nuestro servidor y ejecutaremos el comando:

$ more /etc/hosts

El resultado sería una tabla similar a la siguiente:

```
# @(#)hosts 1.2 Lachman System V STREAMS TCP source
# SCCS IDENTIFICATION
127.0.0.1   localhost
128.128.128.254   std
128.128.129.2 std2_slip
128.128.129.1 std_slip
128.128.127.254   std2
128.128.128.1       ps1
128.128.128.2       ps2
128.128.128.3       ps3
```

Donde en cada línea figuran los nombres de las estaciones, sus direcciones IP y eventualmente lo que se conoce como «Dominio», concepto que se explicará con posterioridad.

Aunque el propio sistema operativo efectúa operaciones de mantenimiento sobre la tabla de hosts, es labor del administrador de la red incluir los nombres de las estaciones junto con sus direcciones IP.

Se recomienda como ejercicio, realizar una inspección en ambos servidores. Puede intuirse la semejanza existente con la tabla local en C:\TUNTCP\HOSTTAB. Sin embargo la tabla de hosts en Unix posee un significado más extenso, puesto que además de ser un modo abreviado para referirse a una dirección IP, muchas aplicaciones hacen uso de esta tabla para verificar la autenticidad de las estaciones conectadas.

10. El protocolo ARP

Puesto que un enlace de datos como Ethernet o Token Ring posee su propio esquema de direccionamiento (a menudo sobre 48 Bits), cada protocolo de red deberá de amoldarse en consecuencia. Una red como Ethernet puede ser usada de forma simultánea por diferentes capas de red (Unix, Novell, Lantasti, WFG etc. pueden coexistir sobre el mismo soporte físico).

En el seno de una Red Local, cuando una trama Ethernet se envía de una máquina a otra, es la dirección de 48 Bits (dirección MAC) quien determina a qué interface físico va destinada. El «driver» de red nunca se preocupa de la dirección IP de destino contenido dentro del datagrama IP.

La especificación ARP (Address Resolution Protocol), contenida en la RFC 826 es quien se encargará de efectuar una

correspondencia dinámica entre la dirección MAC y la dirección IP.

Cada vez que se ejecuta la orden

$ telnet std

se desencadena la siguiente serie de acontecimientos:

• El cliente telnet, llama a una función (gethostbyname) para convertir el nombre de la máquina (std) en una dirección IP de 32 Bits.

• El cliente telnet pide a su TCP utilizar esta dirección para establecer una conexión.

• TCP envía una petición de conexión a la máquina remota emitiendo un datagrama IP a su dirección IP.

• ARP envía una trama especial Ethernet, de tamaño muy corto, llamada **«petición ARP»** a cada máquina de la Red Local. Este proceso se conoce como **«Broadcast»**. La petición ARP contiene la dirección IP de la máquina destino, y equivaldría a formular la pregunta:

«Si Usted es propietario de esta dirección IP, por favor respóndame devolviendo su dirección MAC»

• La capa ARP de la máquina destino recibe el «Broadcast», verifica que el solicitante pide su dirección IP y emite una

«respuesta ARP». Esta respuesta contiene la dirección IP y la dirección MAC correspondiente.

• La respuesta ARP es recibida y el datagrama IP que produjo la petición ARP puede ser emitido.

10.1 La memoria Caché de ARP

El mantenimiento de una memoria caché ARP en cada máquina es esencial para el correcto funcionamiento de ARP. Ello mantiene las correspondencias entre las direcciones IP y las físicas. El plazo normal de expiración de una entrada en la tabla ARP es de 20 minutos después de su creación.

A modo de ejemplo práctico vamos a realizar un «telnet» sobre nuestro servidor, ejecutando el siguiente comando:

$ arp -a

El resultado dependerá de en qué condiciones se encuentre la red. Sería muy probable obtener una serie de líneas similares a las siguientes:

ps22 (128.128.127.22) at 2:60:8c:e:3b:c

ps20 (128.128.127.22) at 2:60:8c:a:60:dd

...

El comando «arp» posee una serie de opciones que permiten efectuar distintos ajustes en la memoria caché de ARP.

arp hostname
Visualiza dirección MAC de <hostname>

arp -a [/unix] [/dev/kmem]
Visualiza toda la tabla

arp -d hostname
Borra dirección MAC de <hostname>

arp -s hostname ether_addr [temp] [pub] [trail]
Añade una nueva entrada

arp -f filename
Añade una nueva tabla desde un fichero

La última opción suele ser útil para optimizar el arranque de servidores con gran carga de estaciones conectadas. Bastaría con introducir el comando en algunas de las «shell» de conexión del sistema.

11. ICMP

ICMP es considerado a menudo como parte de la capa de red IP. Su misión es comunicar los mensajes de error y otras circunstancias que reclaman atención. La RFC 792 contiene la especificación oficial del «Internet Control Message Protocol».

Los mensajes ICMP son transmitidos en el interior de datagramas IP, como lo muestra la figura.

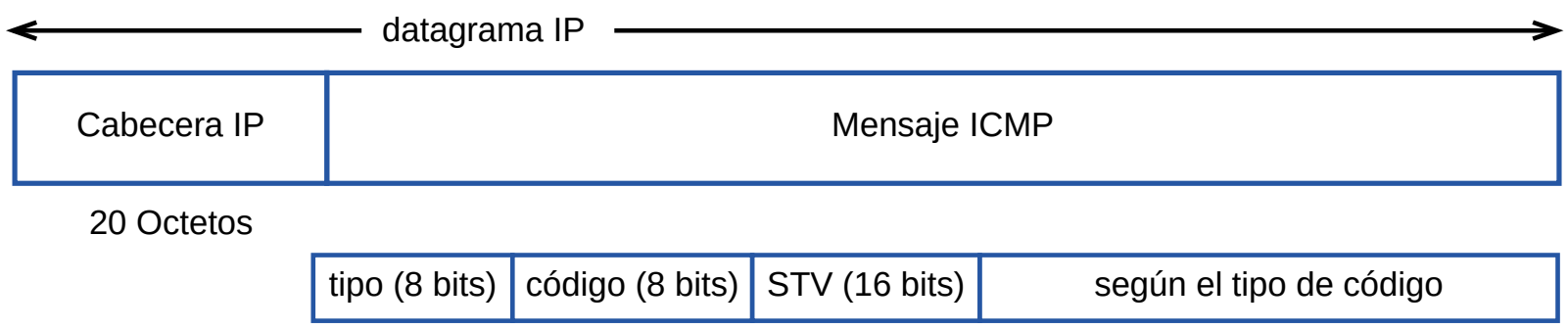


Figura 12

A continuación se describen los distintos **grupos** ICMP, dentro de los cuales existen varios tipos de mensajes:

- *Respuesta eco (Respuesta de Ping).* *Tipo 0*
- *Destino inaccesible.* *Tipo 3*
- *Cadencia de envío demasiado elevada (Control de flujo)* *Tipo 4*
- *Redirección.* *Tipo 5*
- *Petición eco.* *Tipo 8*
- *Aviso de Router.* *Tipo 9*
- *Solicitud de Router.* *Tipo 10*
- *Tiempo sobrepasado (ttl).* *Tipo 11*
- *Problema de parámetros o de forma.* *Tipo 12*
- *Petición «Timestamp»* *Tipo 13*
- *Respuesta «Timestamp»* *Tipo 14*
- *Petición de Información (Obsoleto).* *Tipo 15*
- *Respuesta de Información (Obsoleto).* *Tipo 16*
- *Petición de máscara de dirección.* *Tipo 17*
- *Respuesta de máscara de dirección.* *Tipo 18*

El campo tipo se complementa con la información suministrada por el campo *code*. En función del valor que tomen ambos, el receptor puede distinguir entre una solicitud o un error ICMP.

En el primer caso, se solicita o se informa de un acontecimiento concreto que no es causa de ningún error. Una respuesta a un eco ICMP lanzado por «ping» o un aviso de presencia de Router, serían un ejemplo de ello.

En el segundo caso, se ha producido una condición de error y es necesario emitir una notificación. El campo TTL a cero, una máquina inaccesible, una fragmentación requerida pero imposible de realizar debido al bit «don’t fragment», etc., serían ejemplos de errores ICMP.

Es necesario establecer una distinción entre una solicitud ICMP y un error ICMP. Un mensaje de error, nunca puede ser producido como consecuencia de otro mensaje de error. Si esta regla no fuese aplicada, podríamos encontrarnos sobre escenarios en los que un error ICMP provoca otro error ICMP y así sucesivamente.

Un error ICMP enviado contiene siempre la cabecera IP y los 8 primeros octetos de datos del datagrama que lo provocó. Ello permite al módulo ICMP asociar el mensaje recibido a un protocolo particular (TCP o UDP en función del campo «protocolo» de la cabecera IP) y a un proceso de usuario determinado (mediante los números de puerto de TCP o UDP).

Las situaciones expuestas a continuación no generan mensajes de error ICMP:

1. Un mensaje de error ICMP (Un mensaje de error ICMP puede, a pesar de todo, ser generado como respuesta a una solicitud ICMP).
2. Un datagrama destinado a una dirección IP de «broadcast».
3. Un datagrama enviado como «broadcast» de la capa de enlace.
4. Un fragmento recibido fuera de secuencia.
5. Un datagrama cuya dirección fuente no ha sido emitida por una única máquina. Esto significa que la dirección fuente no puede valer 0, ni ser el bucle local, ni una dirección broadcast.

Estas reglas son necesarias para prevenir las «tormentas de broadcast» *(broadcast storm)*, que aparecían en el pasado cuando los errores ICMP se generaban como respuesta a paquetes emitidos bajo la forma de «broadcast».

12. Encaminamiento IP

El encaminamiento es una de las funciones más importantes del protocolo IP. Los datagramas pueden ser emitidos hacia una máquina local o hacia una máquina remota. En el último caso, ésta máquina deberá ser configurada como un **Router, Gateway** o **Pasarela**, o los datagramas recibidos que no pertenezcan a la máquina serán destruidos en silencio.

Los procesos más habituales o «daemons» que implementan estas funciones en Unix son **routed** y **gated**.

La elección de un protocolo de encaminamiento en función de una determinada máquina, los modos de intercambio de información de trayectorias con los Routers adyacentes y el funcionamiento de los protocolos de encaminamiento, son temas complejos y muy extensos que no vamos a abordar.

Nuestra preocupación consistirá en cómo una simple capa IP toma sus decisiones de encaminamiento.

La secuencia mediante la cual IP a partir de la tabla de encaminamiento decide por dónde enviará un determinado datagrama es la siguiente:

- *Búsqueda de una dirección completa de máquina correspondiente*
- *Búsqueda de una dirección de red correspondiente*
- *Búsqueda de una entrada por defecto*

Es necesario efectuar una distinción entre **mecanismo de encaminamiento** y **política de encaminamiento**. En el primer caso IP busca en la tabla de rutas y decide a qué interface enviar el datagrama. En el segundo caso un **«daemon»** especializado como **routed** o **gated**, decide qué rutas definir en su tabla para que pueda actuar el mecanismo de encaminamiento. Nos ocuparemos principalmente del primer caso.

12.1 Mecanismo de encaminamiento

Veámoslo a través de un ejemplo. Para ello efectuaremos una solicitud «telnet» hacia nuestro servidor y accederemos como usuario, haciendo uso del comando **netstat** ya utilizado anteriormente pero con opciones diferentes:

$ netstat -rn

Obtendremos la tabla de encaminamiento actual (opción -r), mostrando las direcciones IP con notación decimal (opción -n):

Routing tables

	Destination	Gateway	Flags	Refs	Use	Interface
1)	128.128.129.1	128.128.129.2	UH	1	8	sl0
2)	127.0.0.1	127.0.0.1	UH	1	0	lo0
3)	128.128.129.2	127.0.0.1	UH	1	0	lo0
4)	128.128.128	128.128.129.1	UG	0	0	sl0
5)	default	128.128.129.2	U	0	197	sl0
6)	128.128.127	128.128.127.254	U	4	35	e3A0

Para la correcta interpretación de la tabla, necesitaremos tener a mano el esquema de la topología de la red con las direcciones IP de cada máquina así como la máscara de subred empleada.

En el campo **«Flags»** pueden existir 5 tipos diferentes:

U La ruta está en servicio

G La ruta es un Router (Gateway). Si este «flag» no está posicionado, el destino está directamente conectado sobre el Router. Tal es el caso de las entradas 1,2,3,5 y 6.

H La ruta hace referencia a otra máquina, el destino es una dirección de máquina completa. La no existencia de este indicador implica que la ruta incluye otra red, y el destino es una dirección de red: Identificador de red o de Subred.

D La ruta ha sido creada por una **redirección** (Mecanismo que se activa durante la emisión de un datagrama a un Router cuando debería de haber sido para otro).

M La ruta ha sido modificada por una redirección.

El indicador o «Flag» **G** tiene una especial importancia por cuanto permite distinguir entre una **ruta directa** y **otra indirecta.** La diferencia reside en que un paquete sobre una ruta directa posee a la vez la dirección MAC e IP de la máquina destino. Mientras que un paquete emitido sobre una ruta indirecta posee la dirección IP del destino pero la dirección MAC del próximo Router.

Supongamos que llega un datagrama con la dirección 128.128.129.1. La búsqueda tendrá éxito en la primera entrada y el datagrama será enviado por el interface físico sl0 con dirección 128.128.129.2.

Nótese que las conexiones punto a punto, han de ser definidas de forma absoluta, es decir, especificando la dirección completa de máquina en cada extremo de la conexión.

La entrada número 4, indica una dirección indirecta hacia la red 128.128.128.0. Cualquier datagrama con esas características será retransmitido al Router 128.128.129.1 que pertenece al otro extremo de la conexión SLIP.

Los datagramas enviados sobre el segmento 10Base2 perteneciente a la tarjeta de red e3A0 están definidos por la entrada número 6 donde aparece una dirección destino de red, la 128.128.127. y el propio gateway 128.128.127.254 .

Todas aquellas direcciones que no pertenezcan a ninguno de los destinos especificados en la tabla, serán reconducidos a la entrada por defecto (5), perteneciente a la conexión SLIP.

La columna «Refs» es un contador que indica el número de veces que una ruta es utilizada. Puesto que TCP es un protocolo orientado a conexión, una ruta será conservada mientras dure el enlace. Si establecemos un servicio Telnet hacia el servidor, se podrá observar que este contador se incrementa en una unidad.

La columna encabezada con «Use» expresa la cantidad de paquetes enviados a través de esta ruta. Si somos los únicos en utilizarla, y lanzamos el programa Ping con el fin de enviar 5 paquetes, veremos que el contador se incrementa en 5.

De forma esquemática, podría todo quedar resumido como se muestra en la figura 13.

12.2 Creación de Rutas

Pueden ser creadas de forma estática, es decir, simplemente con la ayuda del comando **route**, o de forma dinámica mediante un «daemon» o proceso especializado que en la mayoría de los sistemas Unix suele llamarse **routed** o **gated.**

$ route add default 128.128.129.1 0 (Métrica 0 si local y >0 si externa)

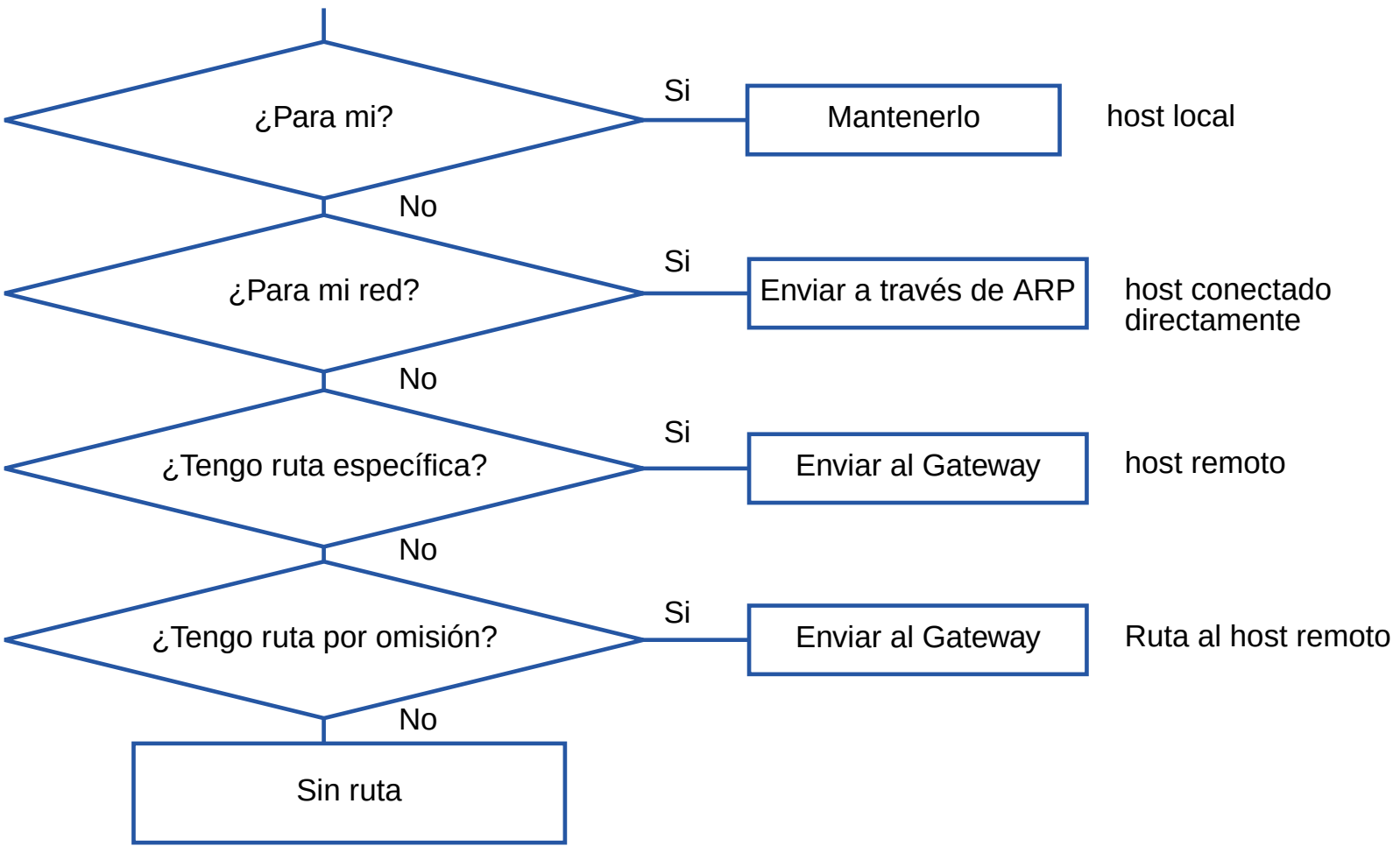


Figura 13

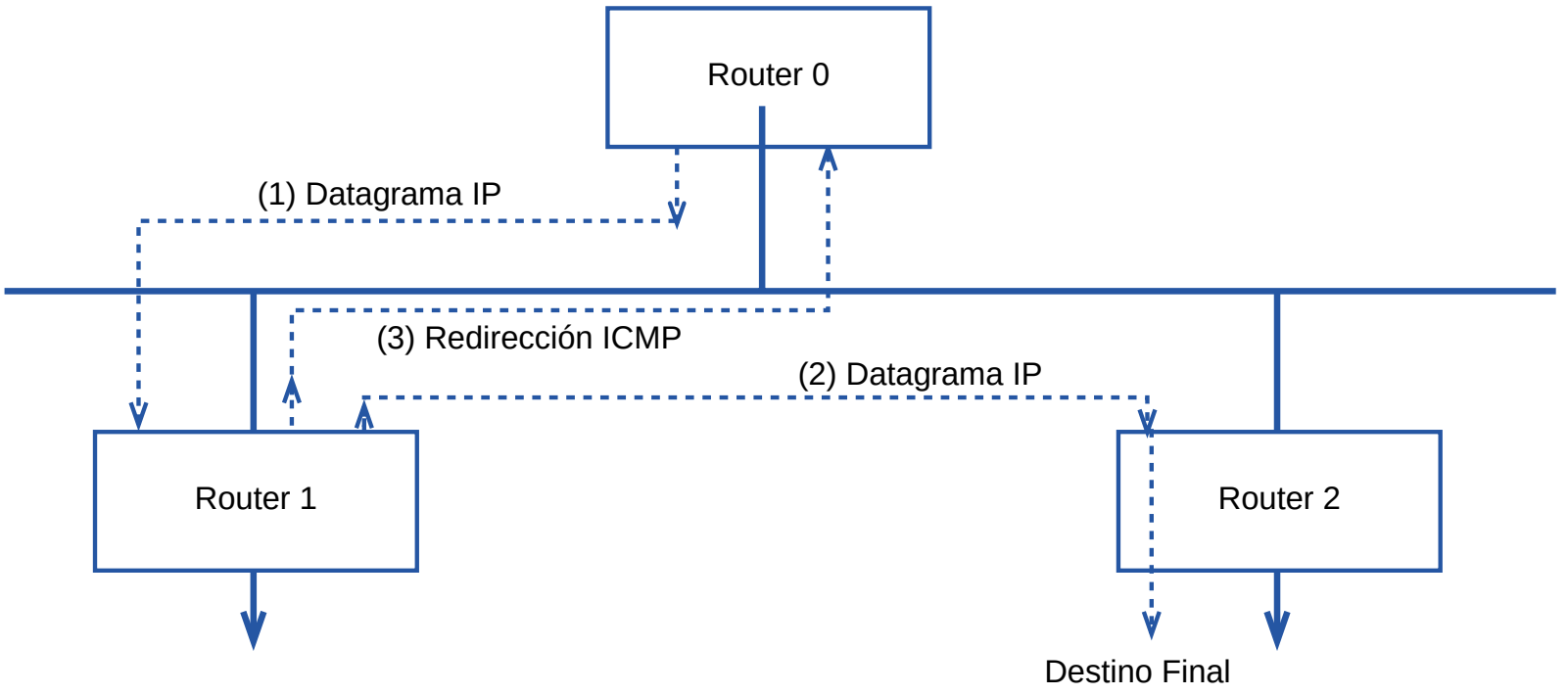


Figura 14

12.3 Error de Redirección ICMP

El error de redirección ICMP es enviado por un Router hacia el emisor de un datagrama IP cuando este datagrama debería de haber sido transmitido a un Router diferente. Dado que para reproducir esta circunstancia se necesitan como mínimo 3 Routers, lo vamos a ver únicamente de forma gráfica.

Siguiendo el esquema de la figura 14, veamos paso a paso qué ocurre en la configuración representada:

1. Suponemos que el Router 0 envía un datagrama IP al Router 1. La decisión de encaminamiento es coherente puesto que el Router 1 está definido por defecto en ésta máquina.

2. Router 1 recibe el datagrama, interroga su tabla de rutas, y determina que Router 2 es el siguiente salto de envío. En el momento de enviarlo, percibe que el interface de salida es el mismo por donde recibió el datagrama procedente de Router 0.

3. Procede por tanto a emitir un **error de redirección ICMP** hacia Router 0, sugiriéndole que actualice su tabla de encaminamiento y que envíe directamente los próximos datagramas a Router 2 sin pasar por Router 1. Simultáneamente, envía el datagrama a Router 2.

12.4 Encaminamiento Dinámico

Hasta ahora únicamente hemos considerado el caso en que una tabla de encaminamiento era creada a través de **«route add»** bien en el inicio o durante un funcionamiento normal. Este tipo de encaminamiento se considera **estático.**

El encaminamiento dinámico pone en marcha un protocolo de comunicación **entre Routers** de forma que cada uno de ellos tiene la suficiente información para actualizar sus tablas de rutas de forma **dinámica.**

Los sistemas Unix ejecutan frecuentemente un proceso o **«daemon»** de encaminamiento denominado **routed,** ya

comentado anteriormente. Se proporciona con casi la totalidad de implementaciones TCP/IP. Este **daemon** utiliza el protocolo denominado **RIP.**

El **proceso gated,** posee un radio de acción mucho más amplio que **routed** y puede utilizar los protocolos **RIP, HELLO** y **EGP.**

HELLO posee una métrica basada en **retardos** de enlaces de la red, permite sincronizar los relojes de un grupo de máquinas así como el cálculo de vías de acceso óptimas que ofrezcan menor retardo.

EGP soporta un mecanismo de adquisición de **gateways** adjuntos permitiendo el intercambio de información de accesibilidad. Cada **gateway** comprueba continuamente si sus **EGP** adjuntos responden.

12.5 Routing Information Protocol (RIP)

RIP utiliza una métrica basada en **número de saltos**, donde el máximo se sitúa en 16. Es por tanto utilizado en redes con dimensiones reducidas en cuanto a número de Routers. En el caso que nos ocupa es éste el que permite la comunicación de encaminamiento entre el servidor **std** y **std2**. Una métrica de 16 indica el valor infinito.

Los mensajes **RIP** son transportados por datagramas **UDP** con el número de puerto 520. Cuando se inicia el proceso, son enviadas solicitudes **RIP** por todos los interfaces activos reclamando las tablas de encaminamiento de los Routers adyacentes.

Un mensaje **RIP** utiliza 20 octetos por ruta (sin contar los 20 de IP y los 8 de UDP), permitiendo señalar un *máximo de 25 rutas por mensaje* y así conservar un tamaño no superior a 512 bytes por mensaje (20x25 + 4 de cabecera = 504), es decir 512 bytes por datagrama UDP.

Cada 30 segundos, una parte o la totalidad de la tabla de encaminamiento es enviada a los Routers adyacentes por medio de un *«broadcast»* o al otro lado de una conexión punto a punto.

Por cada ruta existe un temporizador asociado. Un sistema utilizando **RIP,** que encuentra una ruta no actualizada desde 3 minutos, procede a marcarla para su destrucción con el valor infinito (16). Esto significa que han faltado 6 actualizaciones de 30 segundos por parte del Router que comunicó esa ruta. La eliminación permanente se retrasa 60 segundos más para asegurarse de que esta acción ha sido notificada al resto de la red con el tiempo suficiente.

12.6 Comprobando el funcionamiento de RIP

Para verificar el correcto funcionamiento del protocolo **RIP**, puede ser útil hacer uso del comando **ripquery**, cuya finalidad es la de permitir comprobar las actualizaciones periódicas realizadas por los sistemas adyacentes que ejecutan **RIP**.

Por ejemplo, para comprobar si se reciben actualizaciones de los servidores **std2** y **std3,** ejecutaríamos el siguiente comando desde **std:**

$ ripquery –n –r std2 std3

La opción -n indica que se visualicen las direcciones en forma numérica, sin intentar la conversión a nombres.

Con -r solicitamos el uso del comando REQUEST en vez del comando POLL para interrogar al suministrador RIP. El comando POLL no está universalmente soportado, por lo que es más aconsejable utilizar la opción -r.

Esta utilidad permite verificar el funcionamiento de los procesos **RIP**, así como la correcta configuración de las tablas de encaminamiento.

12.6.1 BGP (Border Gateway Protocol)

Aunque BGP, o protocolo de pasarela frontera, se desarrolló para su uso con conjuntos de redes que emplean la arquitectura de protocolos TCP/IP, los conceptos que define son aplicables a cualquier conjunto de redes. BGP se ha convertido en el protocolo de dispositivo de encaminamiento exterior estándar en Internet.

BGP se diseñó para permitir la cooperación en el intercambio de información de encaminamiento entre dispositivos de encaminamiento, llamados pasarelas en el estándar, en sistemas autónomos diferentes. El protocolo opera en términos de mensajes, que se envían utilizando conexiones TCP. Cada mensaje comienza con una cabecera de 19 octetos, que contiene tres campos:

- **Marcador.** Este campo es usado como parte de un mecanismo de autentificación. El emisor puede insertar un valor en él para permitir al destino verificar la identidad del emisor.
- **Longitud** del mensaje, en octetos.
- **Tipo** de mensaje. Cada tipo de mensaje dispone de su propio formato. Estos son los posibles tipos de mensajes:

– *Open.* Este tipo de mensaje sirve para establecer una relación de vecindad con otro dispositivo de encaminamiento.

– *Update*. Sirve para transmitir información sobre una única ruta y/o enumerar rutas múltiples que se van a eliminar.

– *Keepalive.* Sirve para confirmar un mensaje *Open* y confirmar periódicamente la relación de vecindad.

– *Notification.* Se envía cuando se detecta una condición de error.

BGP involucra tres procedimientos funcionales:

1. **Adquisición de vecino.** Si los dos dispositivos de encaminamiento están en sistemas autónomos diferentes, podrían desear intercambiar información de encaminamiento. Para esto, primero se realiza la adquisición de vecino. El término *vecino* se refiere a dos dispositivos de encaminamiento que comparten la misma subred. La adquisición de vecino ocurre cuando dos dispositivos de encaminamiento vecinos en diferentes sistemas autónomos se ponen de acuerdo en intercambiar regularmente información de encaminamiento. Se requiere un procedimiento formal de adquisición ya que uno de los dispositivos de encaminamiento puede no querer participar. Así, un dispositivo envía un mensaje Open al otro, el cual puede aceptar, devolviendo un mensaje *Keepalive,* o rechazar el ofrecimiento.

2. **Detección de vecino alcanzable.** Para mantener la relación, se utiliza el procedimiento de detección de vecino alcanzable, según el cual, ambos dispositivos de encaminamiento se envían periódicamente mensajes *Keepalive.*

3. **Detección de red alcanzable.** Este procedimiento consiste en que cada dispositivo mantiene una base de datos con las subredes que puede alcanzar y la ruta preferida para alcanzar esa subred. Siempre que se realiza un cambio en esta base de datos, el dispositivo de encaminamiento envía un mensaje *Update* por difusión a todos los otros dispositivos de encaminamiento que implementan BGP.

12.7 Registro de Ruta de «Ping»

La mayoría de versiones de «ping» proporcionan la opción **-R** que activa la funcionalidad de registro de ruta. Ping pone entonces en funcionamiento dentro del datagrama IP el modo **«RR»** sobre el datagrama emitido, que a su vez contiene el mensaje de petición **ICMP.**

Esta opción indica a cada **Router** de gestionar el datagrama con el fin de añadir su dirección IP a una lista dentro de un campo denominado «opciones» del mismo datagrama. Cuando éste alcanza su destino, la lista completa de direc-

ciones IP se copia en la respuesta ICMP generada por el destinatario.

Asimismo, todos los **Routers** situados sobre el camino de regreso añaden igualmente sus direcciones IP a la lista.

Cuando «ping» recibe la respuesta ICMP, visualiza una tabla similar a la siguiente:

(desde std2)

$ ping -R 128.128.128.27

PING 128.128.128.27 (128.128.128.27): 56 data bytes

104 bytes from 128.128.128.27: icmp_seq=0 ttl=254 time=270 ms

RR: std (128.128.128.254)

std_slip (128.128.129.1)

localhost (127.0.0.1)

104 bytes from 128.128.128.27: icmp_seq=1 ttl=254 time=240 ms (same route)

104 bytes from 128.128.128.27: icmp_seq=2 ttl=254 time=220 ms (same route)

104 bytes from 128.128.128.27: icmp_seq=3 ttl=254 time=250 ms (same route)

– 128.128.128.27 ping statistics –
4 packets transmitted, 4 packets received, 0% packet loss
round-trip min/avg/max = 220/245/270 ms

La opción **-R** existe en la mayoría de Routers existentes en la actualidad, aunque puede darse la excepción. En tal caso faltaría la anotación correspondiente en la respuesta dada por «ping».

Cuando un **Router** graba su dirección IP sobre la lista ¿Qué dirección usa entre todas las de sus interfaces? La RFC 791 especifica que el Router debe anotar la dirección de su Interfaz de salida.

El datagrama enviado desde la estación **std2** alcanza al Router **std** quien lo encamina por su Interfaz LAN Ethernet (e3A0) con dirección IP 128.128.128.254 (std). Al alcanzar el destino, la lista (en este caso sólo 1 dirección) se copia sobre la respuesta ICMP y se devuelve a **std.** Según su tabla de encaminamiento, corresponde al Interfaz std_slip (128.128.129.1) la retransmisión hacia **std2.**

A partir de aquí ocurre algo insólito: Parece ser el **bucle local (127.0.0.1)** de **std2**, el Interfaz destino de la respuesta ICMP.

Esto es lógico si observamos la tabla de encaminamiento de **std2:**

Routing tables

Destination	Gateway	Flags	Refs	Use	Interface
128.128.129.1	128.128.129.2	UH	1	8	sl0
127.0.0.1	127.0.0.1	UH	1	0	lo0
128.128.129.2	127.0.0.1	UH	1	0	lo0
128.128.128	128.128.129.1	UG	0	0	sl0
128.128.127	128.128.127.254	U	4	35	e3A0

El datagrama de respuesta ICMP posee la dirección destino 128.128.129.2. Observando la tabla, existe una entrada que hace alusión a la dirección de máquina 128.128.129.2, cuyo Gateway destino es el bucle local.

La tabla de encaminamiento, en este caso, ha sido creada íntegramente y de forma automática durante el proceso de arranque por el daemon **routed** a través del protocolo **RIP**.

13. TCP y UDP

13.1 Introducción

Aunque TCP y UDP utilizan la misma **capa de red,** TCP proporciona un servicio totalmente diferente hacia la **capa de aplicación** en comparación con UDP.

Ambos protocolos se sitúan en la capa de transporte y la elección de uno u otro dependerá de las necesidades de la aplicación que lo utilice.

TCP proporciona un servicio de flujo de octetos **orientado a la conexión** y **fiable**. Significa que para establecer el enlace, deben cumplirse unos «formalismos» de protocolo antes de empezar la transmisión de datos. Ello es comparable al sistema telefónico: Si deseamos hablar con alguien, hemos de

marcar primero el número y esperar a oír la voz de nuestro interlocutor, para iniciar la conversación.

Las dos aplicaciones que utilizan TCP (Cliente y Servidor) deben de establecer por tanto una conexión TCP, antes de intercambiar los datos.

TCP aporta fiabilidad realizando las siguientes acciones:

– Los datos de la aplicación son reducidos a fragmentos donde la talla corresponde a la mejor elegida por TCP para la transmisión. Esto es completamente diferente de **UDP** donde cada escritura de la aplicación genera un datagrama **UDP** con ese tamaño. La unidad de información emitida por TCP es conocida como **segmento.**

– Cuando TCP emite un **segmento,** mantiene un temporizador esperando su asentimiento por parte del otro extremo.

– Si TCP recibe datos del otro lado de la conexión, emite un asentimiento.

– TCP mantiene una suma de control su cabecera y sus datos. Si aparece un segmento inválido, se rechaza y no se emite el asentimiento de éste.

– Puesto que los segmentos TCP son emitidos como datagramas IP, y puesto que los datagramas IP pueden llegar en

completo desorden, del mismo modo, los segmentos TCP podrán llegar desordenados. Una recepción TCP reorganiza los datos si es necesario, pasándolos en el orden correcto a la aplicación.

– Como los datagramas IP pueden ser duplicados, TCP elimina siempre los datos duplicados.

– TCP proporciona un control de flujo. Cada extremo de la conexión TCP dispone de un tamaño definido de ventana.

Un flujo de octetos de 8 bits es intercambiado a lo largo de la conexión TCP entre las dos aplicaciones. No existen delimitadores de registro insertados por TCP. Es lo que se conoce como un **servicio de flujo de octetos** *(byte stream service)*. Si una aplicación escribe 10 octetos y luego 20 y después 50 octetos, la aplicación situada al otro extremo puede leer los 80 octetos en cuatro lecturas de 20. Un extremo coloca un flujo de octetos en TCP, y el mismo flujo aparece al otro lado.

TCP no interpreta nunca el contenido de los octetos. Ignora si se trata de datos binarios, ASCII, EBCDIC o cualquier otra cosa. Es decir, es un protocolo transparente. La interpretación de ese flujo de datos corresponde a las aplicaciones situadas a cada extremo.

UDP es un protocolo de transporte sencillo no orientado a la conexión. Cada operación de salida efectuada por un proceso determinado, genera un único datagrama UDP, provocando la emisión de un datagrama IP.

UDP no ofrece ninguna garantía de fiabilidad: envía datagramas que la aplicación emite hacia la capa IP, pero no existe seguridad de que algún día lleguen a su destino. Dada esta falta de fiabilidad, parece lógico pensar que debemos utilizar siempre TCP. Ello no es del todo cierto puesto que existen casos en los que es preferible la utilización de UDP dada su sencillez y bajo consumo de recursos.

Cuando una aplicación utiliza UDP, debe de controlar expresamente el tamaño de los datagramas IP que emite. Si esta talla excede el propio **MTU**, se producirá una **fragmentación** del datagrama IP. Esta operación se aplica en cada red que atraviesa el datagrama desde su origen hasta su destino y no sólo a la máquina emisora.

El **MTU** no tiene porqué ser el mismo en cada red, y se denomina **MTU de camino** al más pequeño de todas las redes atravesadas por el datagrama.

Las sumas de control de TCP, UDP e IP abarcan los siguientes espectros:

UDP. Protege únicamente la cabecera UDP y sus datos

TCP. Protege únicamente la cabecera TCP y sus datos

IP. Abarca sólo la cabecera IP y no tiene en cuenta el datagrama IP completo.

13.2 TCP

Un mecanismo denominado Asentimiento Positivo con Retransmisión (PAR) proporciona la fiabilidad requerida. Los datos son reenviados transcurrido un tiempo si no se recibe una confirmación positiva por parte de la máquina remota.

Si los datos recibidos son correctos, se emite un asentimiento positivo hacia el emisor. En caso contrario se ignoran los datos recibidos, y transcurrido un cierto tiempo, la unidad TCP emisora reenviará cualquier segmento no confirmado correctamente por el receptor.

Como ya se comentó, TCP está orientado a conexión. Establece un conexión lógica extremo a extremo, entre dos máquinas. Para ello se intercambia una información de control previa a la transmisión de datos.

TCP señala funciones de control a través de la activación de determinados bits en el campo «Flags» de su cabecera.

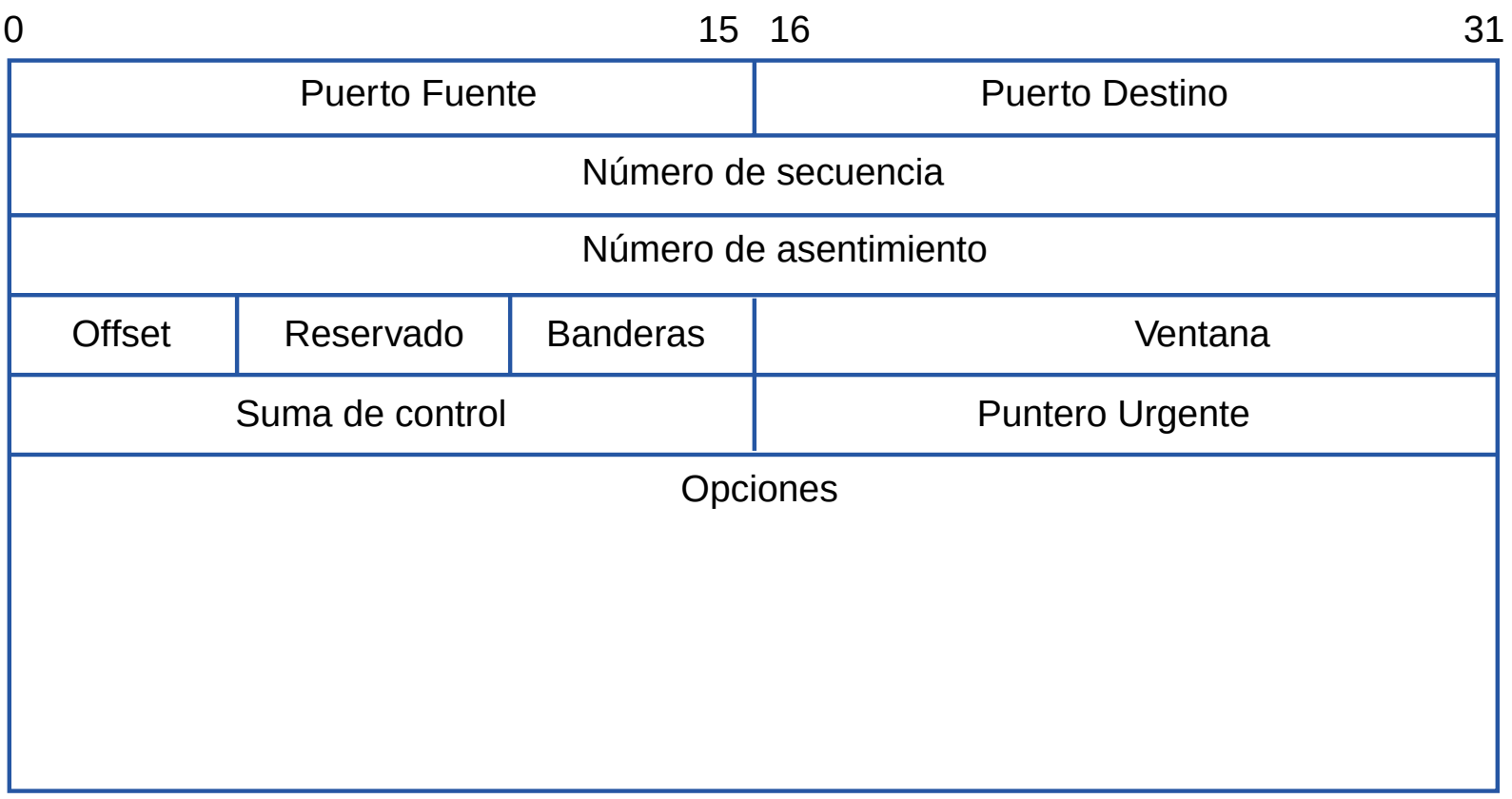


Figura 15

El establecimiento de conexión utilizado por TCP se conoce como «three-way handshake», literalmente «apretón de manos a tres vías», debido a que se necesitan tres segmentos TCP. La figura 16 muestra cómo se lleva a cabo este proceso.

La máquina **A** inicia la conexión enviando a la **B** un segmento con el bit **SYN** activado. La finalidad es comunicar a la máquina **B** la intención de establecer una comunicación TCP, e informarle qué número de secuencia utilizará para iniciar la numeración de todos sus segmentos.

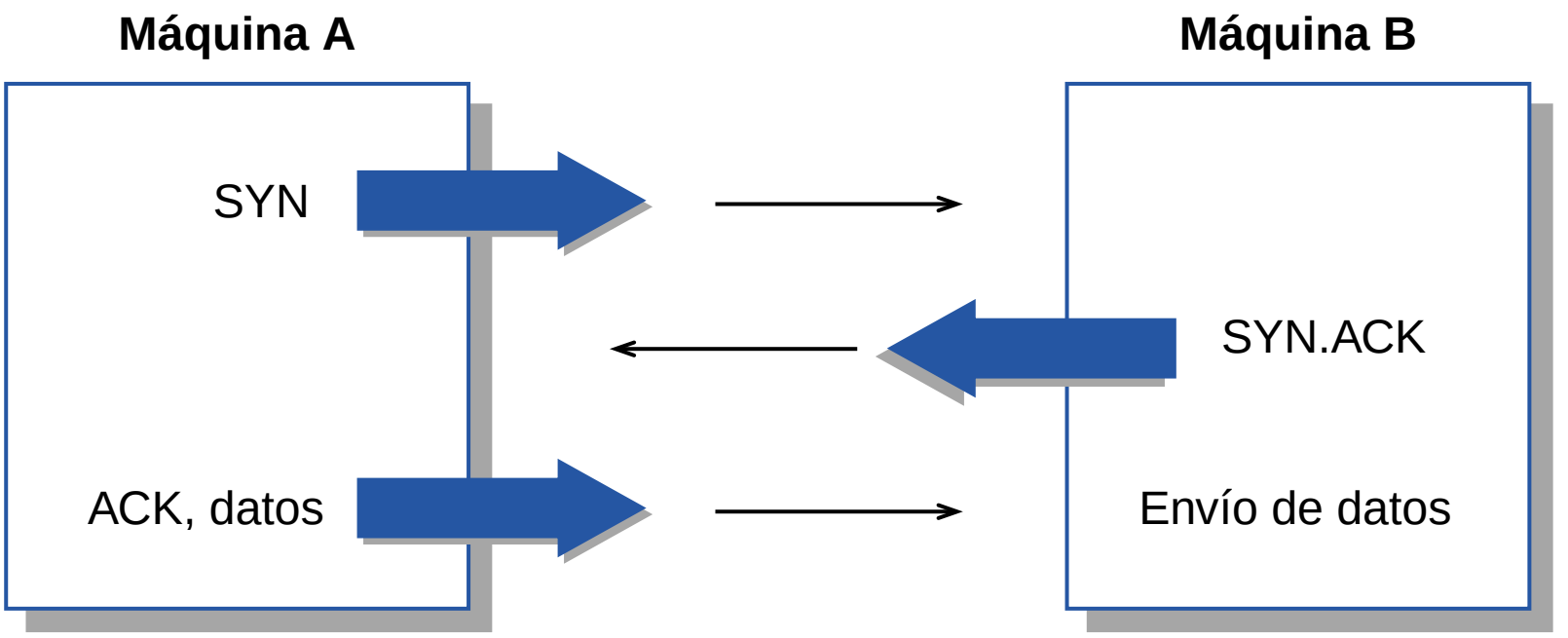


Figura 16

La máquina **B** responde con un asentimiento o **ACK** correspondiente al enviado anteriormente por **A** y con el bit **SYN** activado, indicando así cual será el primer número de secuencia con el que comenzará la numeración de todos sus segmentos.

Por último, la máquina **A** envía un segmento **ACK** confirmando el anterior, e iniciando la transmisión de sus datos.

Después de este intercambio, la capa TCP de la estación **A** tiene claras evidencias de que el TCP remoto está operativo y listo para recibir o transmitir datos. Tan pronto como la conexión sea establecida, puede iniciarse la transmisión.

Así como son necesarios tres segmentos para establecer la conexión, hacen falta cuatro para terminarla. Esto se debe al

cierre incompleto de TCP. Dado que una conexión TCP es de tipo **dúplex**, es decir, los datos pueden desfilar de forma independiente en ambas direcciones, cada dirección a su vez puede ser detenida por separado.

La regla es que cada extremo puede enviar un segmento con el bit **FIN** activado mientras está enviando datos. Cuando un TCP recibe un **FIN**, debe notificar a la aplicación que el otro extremo ha terminado de enviar datos en ese sentido. La emisión de un **FIN**, normalmente suele corresponder a la finalización de una aplicación o una sesión de trabajo.

Por contra, el extremo que ha recibido un **FIN**, puede seguir enviando datos. Esta característica de TCP suele ser utilizada por pocas aplicaciones. El comando Unix **rsh**, es un ejemplo de ello.

El escenario normal de un cierre completo TCP se muestra

en la figura 17.

La capa TCP ve los datos que envía como una corriente de bytes, y no como paquetes independientes.

Por tanto, se necesita mantener la secuencia en la que los bytes son recibidos y enviados. Los campos **«Número de secuencia»** y **«Número de asentimiento»** en la cabecera de

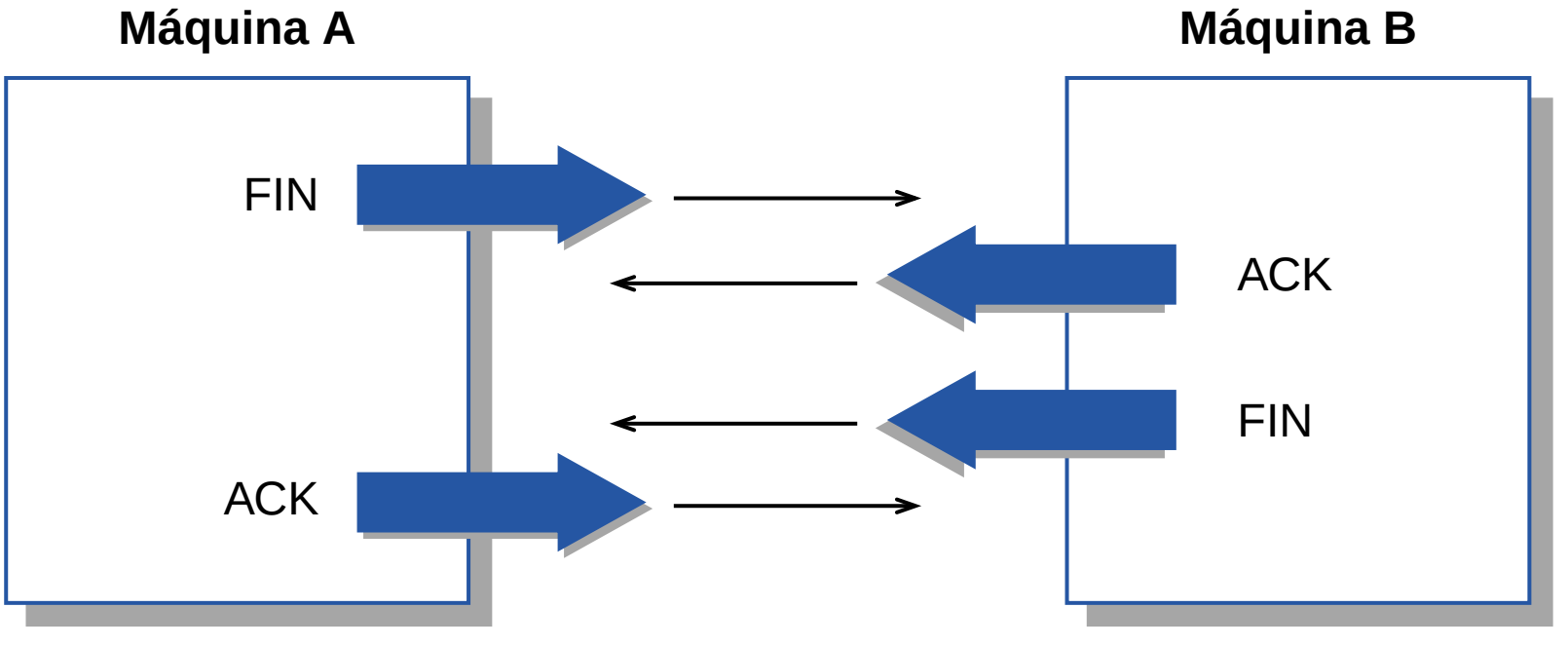


Figura 17

cada segmento TCP son los encargados de llevar a cabo esta función.

La norma no exige que cada sistema comience la numeración de sus bytes con un valor específico. De este modo, antes de iniciar la conexión cada unidad TCP elige el valor inicial o **ISN** (Initial Sequence Number) con el que comenzará su transmisión. Los dos extremos sincronizarán sus **ISN** a través de los segmentos **SYN** utilizados en la fase de establecimiento de la conexión.

Cada byte se numera secuencialmente a partir del **ISN**, de modo que el primer byte real de datos posee el número de secuencia **ISN + 1** (comienza desde cero). El **ISN** situado en la cabecera de cada segmento TCP identifica la posición

secuencial en la corriente de datos del primer byte de datos del segmento. Por ejemplo, si el primer byte en la corriente de datos tiene el número de secuencia 1 y ya han sido transferidos 4000 bytes, entonces el primer byte en el segmento siguiente será el 4001, y el número de secuencia será 4001.

El segmento **ACK** realiza dos funciones: Asentimiento positivo y control de flujo. En el primer caso el receptor notifica al emisor cuántos bytes han sido recibidos, y cuántos puede recibir aún. El número de asentimiento es el número de secuencia del último byte recibido por el extremo remoto. La norma TCP no requiere un asentimiento individual para cada paquete.

El campo **ventana** contiene el número de bytes que el extremo remoto es capaz de aceptar. Si el receptor puede recibir 6000 bytes más, por ejemplo, el valor de la ventana será de 6000. La ventana indica al emisor que puede continuar enviando segmentos mientras que el número de bytes enviados no supere la cantidad que el receptor puede asimilar. El receptor, por tanto, controla el flujo de bytes del emisor cambiando el tamaño de la ventana. Una ventana de cero significa un cese de la transmisión hasta la recepción de un nuevo valor de ventana superior a cero.

TCP es también responsable de la entrega de los datos a la aplicación adecuada. Para ello, las aplicaciones identifican las conexiones TCP a través de los números de puerto, compuestos de 16 bits. Cada segmento queda por consiguiente identificado por los valores de los números de puerto fuente y destino.

13.3 UDP

El protocolo de datagramas de usuario, **UDP,** proporciona a los programas de aplicación un acceso directo a un servicio de entrega de datagramas. Esto permite el intercambio de mensajes entre aplicaciones sobre la red con un gasto mínimo de información redundante debida al protocolo.

UDP es un protocolo no fiable y no orientado a la conexión. Por «no fiable» se entiende que no dispone de mecanismos capaces de recuperar errores, controlar el flujo, ordenar las secuencias de bytes, etc.

Existen sin embargo buenas razones para utilizar UDP en determinadas situaciones.

• Cuando la cantidad de datos a ser transmitidos es muy pequeña, la sobrecarga que supone la creación de conexio-

nes y el asegurar su fiabilidad puede ser muy superior al trabajo de volver a transmitirlos.

• En el caso de tráfico isócrono (sensible al retardo), especialmente la voz o el vídeo, no importa la pérdida de algunos paquetes, mientras que los que lleguen lo hagan en su justo momento. La voz o la imagen pueden degradarse, pero la función global seguirá siendo correcta.

• Funciones de comprobación. Si por ejemplo deseamos diseñar un programa que permita evaluar el **BER** o tasa de error de una conexión, enviaremos datagramas **UDP** para comprobar cuántos han llegado mal. TCP al ser fiable no permitiría conocer los errores puesto que intentaría recuperarlos por si mismo.

• Aplicaciones basadas en el modelo **Petición - Respuesta.** La respuesta puede ser usada como asentimiento positivo a la petición. Si no se recibe respuesta en un intervalo determinado se envía de nuevo la petición.

• Aquellas aplicaciones diseñadas para ser ubicadas en un espacio de memoria relativamente pequeño, y que permiten la carga de otras más complejas. Tal es el caso del protocolo **TFTP** (Protocolo Trivial de Transferencia de Ficheros). Estas implementaciones pueden ser contenidas en una memoria de

tipo EPROM o FLASH, puesto que únicamente necesitan las capas **IP** y **UDP**. Su uso está destinado especialmente a sistemas sin disco, que cargan en memoria principal las aplicaciones a través de la red.

En la figura 18 se puede observar la simplicidad de los 8 bytes que componen la cabecera **UDP**.

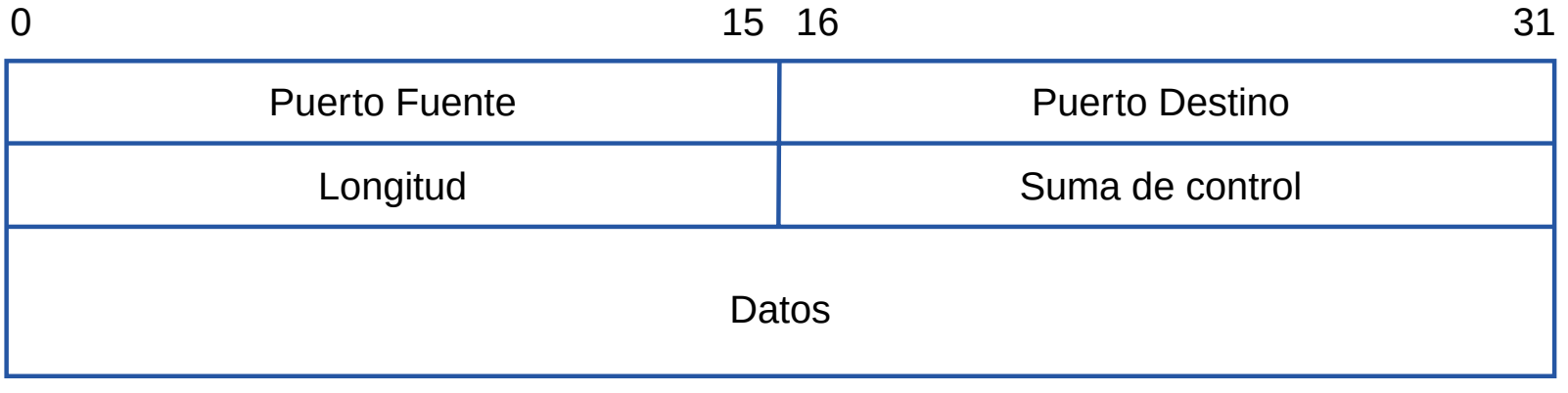


Figura 18

14. Cuestionario 2

1º) Obtener la tabla completa de encaminamiento de las máquinas (A), (B) y (C) respectivamente, correspondientes a la red 140.252. representada en la figura 18.1.

2º) Diferencias entre los protocolos de transporte TCP y UDP

3º) Si en un datagrama IP que transporta UDP, se produce un error en un byte de datos de la aplicación (porción de datos de UDP). ¿Qué protocolo lo detectará primero: IP ó UDP?

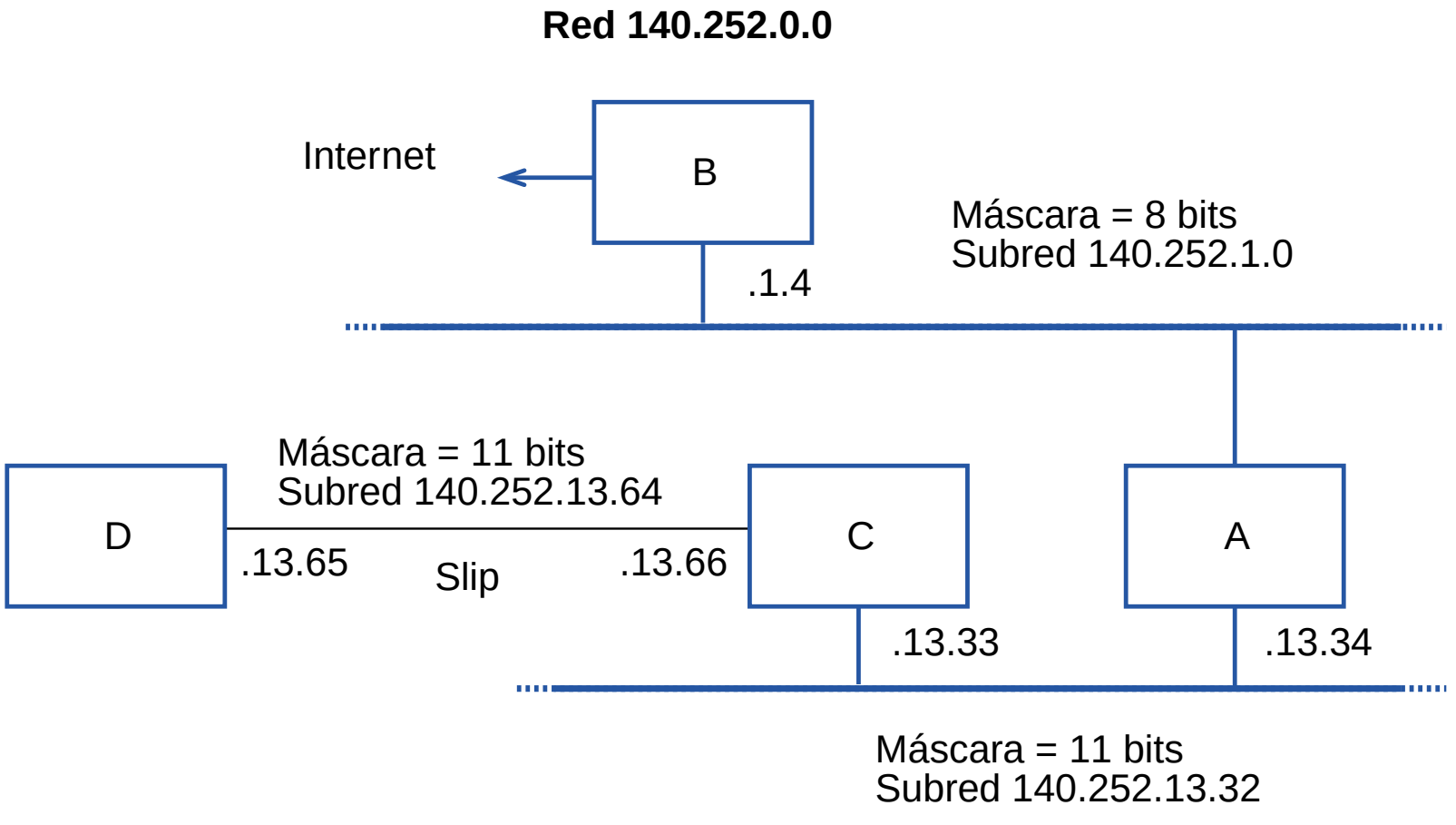

Red 140.252.0.0
Internet
B
.1.4
Máscara = 8 bits
Subred 140.252.1.0
Máscara = 11 bits
Subred 140.252.13.64
D
.13.65
Slip
.13.66
C
A
.13.33
.13.34
Máscara = 11 bits
Subred 140.252.13.32

Figura 18.1

15. Protocolos de Aplicación en TCP/IP

En TCP/IP no hay niveles de sesión ni de presentación. En su día se pensó que no eran necesarios y no se consideraron. En realidad estos son unos niveles muy poco utilizados dentro de la arquitectura OSI.

Por encima de los niveles de sesión y transporte está el nivel de aplicación con los protocolos de más alto nivel. Los protocolos de este nivel de TCP/IP más conocidos y más antiguos son el FTP para la transferencia de archivos, el SMTP para correo electrónico y TELNET para terminal virtual. Pero además existen otros protocolos algo menos conocidos, los cuales se fueron añadiendo a lo largo del tiempo, como son el DNS para nombres de dominio o el HTTP para transferencia de páginas de hipertexto de paginas *web*.

Algunos de los protocolos de aplicación se apoyarán sobre TCP y otros sobre UDP en función de sus necesidades.

En los capítulos siguientes se describirán algunos los protocolos más importantes del nivel de aplicación de TCP/IP.

16. Protocolo de aplicación sobre TCP: TELNET

El protocolo TELNET permite al usuario comunicarse con el S.O. de una máquina remota, iniciar una sesión en ese S.O. y ejecutar programas en la máquina remota, de forma que puede interaccionar con ellos como si la terminal del usuario estuviera conectada directamente a la máquina remota.

Para ello TELNET define el concepto de terminal virtual de red (NVT, *network virtual terminal*) que especifica cómo interaccionan dos terminales utilizando un flujo de caracteres bidireccional. Este flujo de caracteres bidireccional se consigue utilizando una conexión TCP.

La especificación de TELNET permite que los dos puntos extremos puedan negociar un conjunto de funcionalidades opcionales más allá del suministrado por la especificación

básica. La mayor parte del propio protocolo TELNET se dedica a la definición del protocolo de negociación de opciones.

TELNET utiliza el concepto de NTV como apoyo para modelar la conexión entre dos entidades TELNET. Conceptualmente, ambos extremos de una conexión TELNET utilizan un NVT, aunque en las aplicaciones reales, el extremo de la conexión formado por el servidor normalmente es un proceso de identificación (*login*) en lugar de una aplicación de emulación de un terminal TELNET.

El NVT es un dispositivo de caracteres bidireccional. Conceptualmente, consiste en un teclado y una impresora o monitor. Los caracteres procedentes del NVT remoto se imprimen en la impresora local para que el usuario pueda verlos. Los caracteres introducidos con el teclado se envían al NVT remoto y, opcionalmente, se imprimen en la impresora local para proporcionar un eco local. El eco local puede desactivarse mediante la negociación de opciones y, en su lugar, puede ser suministrado por el proceso NVT remoto.

El NVT no tiene especificados ni un ancho ni una longitud de página. Es capaz de reproducir los códigos de caracteres US-ASCII de 7 bits (bit 7 a 0) imprimibles (los códigos del 32 al 126). De los códigos US-ASCII de 7 bits de control sólo tie-

nen significado para el NVT unos pocos. Además de estos códigos de control estándar, la especificación de TELNET define comandos genéricos adicionales que no son códigos de carácter asignados en el conjunto de caracteres ASCII, sino que se señalizan utilizando el protocolo de TELNET, y se definen como datos de 8 bits con el bit 7 activo. Los comandos adicionales de TELNET son los siguientes:

- **IAC (Interpret as command)** (255). Interpretar el siguiente byte como un comando.
- **NOP** (241). Sin operación.
- **EC (Erase Character)** (247). Solicita al NTV remoto que borre el último carácter del flujo de datos.
- **EL (Erase Line)** (248). Solicita al NTV remoto que borre la último línea del flujo de datos (hasta la última secuencia CR-LF).
- **AYT (Are you there)** (246). Solicita al NTV remoto que envíe alguna cadena imprimible como prueba de que aún sigue conectado (normalmente la cadena «yes»):
- **IP (Interrupt Process)** (244). Interrumpe o finaliza el proceso remoto.

- **AO (Abort Output)** (245). Solicita que continúe el proceso remoto hasta acabar, pero sin producir salida.
- **SB (String begin)** (250). Indica que a continuación viene la cadena de negociación de una opción deseada.
- **SE (String end)** (240). Indica que se acaba la cadena de negociación de una opción deseada.
- **WILL** (251). Acordar o solicitar una opción.
- **WON'T** (251). Rehusar una solicitud de opción.
- **DO** (253). Aceptar una solicitud de opción.
- **DON'T** (254). Rehusar aceptar una solicitud de opción.

Por ejemplo, para solicitar al receptor que acepte el modo binario de 8 bits (en vez de la transferencia ASCII de 7 bits habitual) se debe enviar esta cadena de bytes:

IAC, SB, WILL, «0», SE

Donde el «0» representa la opción de negociar el cambio a modo de 8 bits. El receptor puede devolver

IAC, SB, DO, «0», SE

si está dispuesto a aceptar la opción, o

IAC, SB, DON'T, «0», SE

en caso contrario. El receptor también puede iniciar el cambio al modo binario enviando

IAC, SB, DO, «0», SE

Y el transmisor puede devolver

IAC, SB, WILL, «0», SE

si acepta el cambio, o

IAC, SB, WON'T, «0», SE

si no lo acepta.

El puerto asociado a un servidor de TELNET es habitualmente el 23.

TELNET no se utiliza sólo para simples aplicaciones con terminales. Por ejemplo, como se comenta en el apartado sobre FTP, la conexión de control de una sesión FTP utiliza el protocolo TELNET para permitir al cliente y al servidor intercambiar comandos y respuestas FTP. Además, muchos otros protocolos emplean el concepto básico de NVT.

17. Protocolo de aplicación sobre TCP: FTP

El FTP, o protocolo de transferencia de archivos, permite al usuario de una terminal tener acceso a un sistema de archivos remoto e interaccionar con él a través de comandos. Con FTP se puede transferir ficheros de texto o ficheros binarios.

FTP también proporciona características para controlar el acceso de los usuarios. Cuando un usuario solicita la transferencia de un fichero, FTP establece una conexión TCP con el sistema destino para el intercambio de mensajes de control. A través de esta conexión se puede transmitir el identificador y la contraseña del usuario, y el usuario puede especificar el fichero y las acciones sobre él deseadas.

Una vez que se aprueba la transferencia del fichero, se establece una segunda conexión TCP para la transferencia de datos. El fichero se transmite a través de esa conexión de

datos, sin información suplementaria o cualquier cabecera de la capa de aplicación. Cuando la transferencia se completa, el control de la conexión se utiliza para indicar el final y la posibilidad de aceptar nuevas órdenes de transferencia.

Como en otros muchos protocolos de Internet, FTP emplea un sencillo protocolo tipo comando - respuesta. Los comandos FTP son breves cadenas ASCII seguidas de parámetros opcionales dependientes del comando. Estos son algunos comandos FTP:

- USER. Especifica nombre de usuario para el control de acceso.
- PASS. Especifica contraseña de usuario, si se dispone de ella, para el control de acceso.
- CWD. Cambia de directorio de trabajo al nuevo directorio del sistema remoto especificado.
- CDUP. Cambia al directorio padre del directorio de trabajo actual.

El cliente FTP conecta con el servidor en un número de puerto conocido (habitualmente el 21). Esta conexión inicial se convierte en la conexión de control FTP y se utiliza para enviar órdenes y respuestas. La transferencia de los datos tiene lugar en una segunda conexión, en la conexión de

datos. Esta última puede establecerse en el puerto 20 por defecto, pero normalmente se cambia de puerto con los comandos PORT o PASV.

La conexión de control requiere del protocolo TELNET para intercambiar comandos y respuestas orientados a líneas. Si bien TELNET es todo un protocolo por sí mismo, FTP sólo necesita un subconjunto de las funciones de TELNET. La razón principal de utilizar TELNET en la conexión de control reside en que sirve para definir un conjunto básico de caracteres (US-ASCII) y una convención de carácter de fin de línea (CR-LF). Estos son los parámetros por defecto del terminal virtual de red (NVT).

FTP permite especificar la estructura del archivo a transferir, así como su tipo de datos. La aplicación cliente es la responsable de informar al servidor FTP sobre el tipo de archivo y de datos que ha de usar para la transferencia de los mismos. Los tres posibles tipos de estructuras de archivo son éstas:

- **Archivo no estructurado.** Contiene datos binarios o de texto, que se transfieren de forma transparente al nivel de aplicación. Debe ser el usuario quien interprete los datos.

• **Archivo no estructurado.** Está ordenado como una secuencia de registros de tamaño fijo y de tipo definido, que se suelen transferir como bloques de tamaño fijo.

• **Archivos de acceso aleatorio.** Se componen de registros de tamaño variable llamados páginas. Cada página tiene su propia cabecera con información sobre su longitud y sobre la posición de la página en el archivo completo.

Para el tipo de datos del archivo hay cuatro opciones, que se pueden especificar con el comando TYPE:

- Texto **ASCII.**
- Texto **EBCDIC.**
- **IMAGE**, o datos binarios de ocho bits.
- **LOCAL,** o datos binario de tamaño variable.

En nuestro sistema físico, disponemos del servicio **FTP** (ftp.exe) en todas las estaciones o PC de la red y en cada uno de los servidores.

Ello quiere decir que podemos hacer uso del servicio **Telnet** (tnvt52.exe) para conectarnos sobre un servidor y luego lanzar desde allí **«ftp»** (unix) hacia otro servidor diferente.

17.1 FTP desde MS-DOS a través de TUN

Lógicamente, debe existir un camino de conexión TCP/IP hacia la máquina sobre la que queremos conectar, para ello, previamente deberemos de lanzar un **«ping»** de comprobación.

Para activar **FTP** desde MS-DOS se necesita efectuar:

C:\TUNTCP\FTP

Ello provoca la entrada sobre un intérprete de comandos, desde el que podemos comenzar nuestra sesión.

Podemos, sin embargo, utilizar la forma inmediata especificando una serie de parámetros del modo siguiente:

C:\TUNTCP\FTP [-d] [-u usuario password] [-p port] [host] [comando]

donde:

-d Activa el modo «debug».

-u Permite introducir **login** y **password**

-p Especifica el puerto tcp usado (21 por defecto)

host Servidor al que solicitamos el servicio

comando Comando reconocido por el intérprete de comandos de **FTP**, una vez establecida la conexión.

Veamos a continuación algunos de los comandos más relevantes:

! Permite realizar una «shell» al MSDOS

? Visualiza la lista completa de comandos

help [comando] Proporciona información acerca de un comando

aget Inicia una transferencia en modo ASCII desde el host hacia el PC.

aget fichero_remoto fichero_local

aput Inicia una transferencia en modo ASCII desde el PC hacia el host.

aput fichero_remoto fichero_local

bget Inicia una transferencia en modo Binario desde el host hacia el PC.

bget fichero_remoto fichero_local

bput Inicia una transferencia en modo Binario desde el PC hacia el host.

bput fichero_remoto fichero_local

bye Finaliza una sesión y abandona ftp.

cd <camino> Permite cambiar de directorio en la máquina remota. Los caminos se deben especificar en formato Unix.

delete <fichero> Borra un fichero en el host remoto.

dir [camino] Visualiza el contenido de un directorio en la máquina remota.

drive <unidad> Utilizado para cambiar de unidad de disco en la máquina local.

fpwd Indica el nombre del directorio actual en la máquina remota.

lpwd Indica el nombre del directorio actual en la máquina local.

lcd<camino> Cambia el directorio actual en la máquina local.

ldir[camino] Lista el contenido del directorio actual en la máquina local.

mkdir<camino> Crea un directorio en el host remoto.

rmdir<camino> Borra un directorio en el host remoto.

rename Permite cambiar de nombre a un fichero remoto.

rename <fichero1> <fichero2>

ascii Activa el modo de transferencia por defecto a texto. Cuando se transfiere en modo ASCII, existe una conversión de los caracteres <CR> y <LF> para compatibilizar las diferencias existentes entre los ficheros de texto DOS y UNIX.

binary Activa el modo de transferencia por defecto a binario. Tanto ASCII como BINARY, seleccionan el tipo de transferencia utilizada por los comandos **MPUT** y **MGET**.

mget Inicia una transferencia con el modo en curso, desde el host hacia el PC. Permite el uso de caracteres comodín (* ?).

bget fichero_remoto fichero_local

mput Inicia una transferencia con el modo en curso desde el PC hacia el host. Permite el uso de caracteres comodín (* ?).

bput fichero_remoto fichero_local

stat Visualiza el estado de la máquina remota.

El servicio **FTP** cliente reside, al igual que **NFS**, también en los servidores. Por tanto puede ser utilizado a través de una conexión **Telnet** (TNVT52.EXE) sobre cualquiera de ellos, permitiendo la transferencia de archivos entre las máquinas Unix.

Los comandos proporcionados por **FTP** desde Unix son similares en la mayoría de los casos, con algunas diferencias. Lo

correcto sería verificar mediante el comando help la lista de opciones disponibles y actuar en consecuencia. La forma de invocarlo es similar a MS-DOS:

$ ftp std

FTP puede ser automatizado con la ayuda del redireccionamiento. Es decir, si creamos por ejemplo un fichero llamado «auto» mediante un editor de texto, con una serie de comandos **ftp** tales que:

BINARY

CD /u0/ps22

MGET factura0*.*

BYE

y lanzamos desde la línea de comandos del DOS:

C:>\tuntcp\ftp -u ps22 aixftursop std < auto

Provocaremos la transferencia automática de todos los archivos que empiecen por «factura0» del directorio remoto /u0/ps22, hacia nuestra estación de trabajo.

18. Protocolo de aplicación sobre TCP: POP3

POP3 es un protocolo sencillo y muy extendido que permite a una máquina cliente recuperar mensajes de correo electrónico de un servidor. El protocolo **IMAP4** ofrece el mismo servicio pero, además, incorpora el movimiento bidireccional de mensajes y permite la gestión de buzones remotos.

POP3 utiliza el puerto 110 para atender peticiones de conexión TCP.

POP3 está orientado a líneas y es un protocolo de respuesta a peticiones basado en ASCII. El cliente envía los comandos al servidor, el cual devuelve respuestas al cliente. Los comandos POP3 se componen de breves palabras clave, seguidas de parámetros opcionales enviados como una sola línea de texto, seguida de CR-LF. El protocolo emplea únicamente un reducido número de comandos.

Las respuestas al POP3 pueden adoptar dos formas: respuestas de **una sola línea** y respuestas **multi-línea**. Las respuestas de una sola línea indican primero el éxito o el fallo del comando y, luego, suministra información adicional legible por el usuario o adecuada para que las máquinas la analicen. Los códigos básicos POP3 de éxito o fallo son **+OK** y **-ERR**. Cualquier información adicional que aparezca en la línea que sigue a los códigos básicos es descrita con el comando apropiado.

Las respuestas multi-línea consisten en una respuesta de una sola línea seguida de líneas adicionales de información adecuadas al comando que invocó la respuesta. Una respuesta multi-línea finaliza con una línea que contiene un solo carácter de punto seguido de CR-LF. Esta línea final no se considera parte de la respuesta. Cualquier línea de la respuesta multi-línea que comience por un punto lleva un punto adicional insertado antes del primer carácter. Esto asegura que el cliente no las confunda con la línea de finalización. El cliente elimina el punto inicial de todas las líneas que no son la línea de finalización. Este proceso se denomina *dot stuffing* (relleno con puntos).

19. Protocolo de aplicación sobre TCP: SMTP

El SMTP, o protocolo simple de transferencia de correo, ofrece un servicio de transferencia de correo electrónico (e-mail) desde el sistema de correo de un computador servidor o anfitrión al computador del usuario destinatario (computador local). SMTP no se encarga de aceptar correo de otros usuarios locales, ni de eliminar el correo recibido a su destinatario, y éstas son funciones que se dejan al sistema de correo local, también llamado sistema de correo nativo.

Considerando lo anterior, SMTP queda oculto a las transferencias locales de correo del computador del usuario, y sólo se ejecuta cuando hay que enviar correo a una máquina remota o se recibe correo desde una máquina remota. En el computador de un usuario, se dispone de un servidor de SMTP encargado de recibir correo remoto, un cliente encar-

gado de enviar correo remoto y de unas colas a través de las cuales se comunica SMTP y el Interfaz del sistema de correo local.

Dentro del sistema de correo local se mantiene un buzón por cada usuario, donde se puede depositar o recibir correo. Cada buzón tiene un nombre único compuesto de dos partes. La primera parte, o parte local, es un nombre de usuario único dentro del sistema local. La segunda, o parte global, es el nombre del computador servidor, que debe ser único dentro de toda la internet donde actúa el sistema. La parte global se divide normalmente en varios campos dependientes de la localización del servidor. Un ejemplo de nombre de buzón puede ser usuario@disc.ua.es.

El formato de los mensajes de correo de SMTP consta de una cabecera y del cuerpo del mensaje. Ambas partes se componen de varias líneas de texto ASCII. Una línea en blanco separa la cabecera del cuerpo. Cada línea de la cabecera consta de una palabra clave seguida del carácter de *dos puntos* y una cadena de texto. Hay palabras clave obligatorias y otras opcionales. SMTP no especifica la forma en la que se crean los mensajes, se requiere un programa de correo electrónico nativo o un editor local.

La cabecera mínima de un mensaje es ésta:

TO: nombre del destinatario
FROM: nombre del remitente

Otra cabecera puede ser:

TO: nombre del destinatario
REPLY TO: nombre al que debe enviarse la contestación

Y éste es un ejemplo de cabecera con más campos:

TO: nombre del destinatario
FROM: nombre del remitente
CC: copias para...
SUBJECT: resumen del asunto del mensaje
DATE: fecha del mensaje
ENCRYPTED: indicador de que el cuerpo del mensaje esta cifrado

Otro posible campo que puede aparecer en la cabecera es éste:

RECEIVED FROM: identidad de pasarela

Este campo es añadido por las pasarelas que se encuentran en la ruta que recorre el mensaje, si es que el mensaje pasa por varias redes. Así se puede conocer la ruta que ha seguido un mensaje.

Cuando es un computador local se ha creado ya un mensaje con el formato indicado, el sistema de correo local determina

si debe depositarlo en un buzón local o debe colocarlo en la cola hacia SMTP para su reenvío. SMTP determinará la dirección IP del destinatario según el sistema de nombres de dominio (o DNS; este protocolo se estudiará más adelante), y junto con la dirección de puerto del servidor SMTP al que está conectado el usuario destinatario (normalmente 25) se establece una conexión TCP por la que se transfiere el mensaje de correo.

En la transferencia se intercambian una serie de comandos y respuestas como PDUs de SMTP, todas ellas codificadas como cadenas ASCII acabadas en CR-LF. Cada comando es un simple nombre de comando seguido de ciertos parámetros, dependiendo del comando. Ninguno de los comandos es sensible a las mayúsculas o minúsculas, aunque la información transmitida como parámetros sí puede serlo (como son los nombres de buzones de correo).

Las respuestas consisten en un código numérico de tres dígitos seguido de una cadena que explica la respuesta. El software cliente procesa fácilmente el código numérico y la cadena de error puede pasar para poder ser interpretada por una persona, si así se desea.

Justo después de contactar con el servidor, el cliente espera recibir un simple mensaje de saludo del servidor. Cuando el

cliente recibe el mensaje, envía un comando HELO identificándose a sí mismo. Los siguientes comandos identifican a los receptores de un mensaje y transfieren los propios datos del mensaje.

SMTP es capaz de transferir múltiples mensajes durante una determinada sesión. Cada mensaje puede direccionarse independientemente y no necesita estar relacionado con los otros mensajes enviados durante la sesión. Esta capacidad permite a un cliente SMTP intercambiar un conjunto de mensajes en un solo lote con el servidor SMTP, lo que da como resultado una comunicación más eficaz.

En la práctica hay muchas redes que utilizan protocolos distintos del SMTP para manejar el correo, y resulta necesario el uso de pasarelas de correo para el intercambio de correo. Por ejemplo, una pasarela TCP/IP - OSI, que transfiera correo entre un sistema TCP/IP con SMTP y un sistema de niveles OSI con el protocolo MOTIS para correo electrónico.

20. Protocolos de aplicación sobre UDP: TFTP

Como FTP está diseñado para manejar diversos tipos de archivo, e incluso algoritmos de compresión, resultar ser un protocolo demasiado complejo para aplicaciones de redes de área local. Para este caso se ha definido un protocolo de transferencia trivial de archivos (TFTP) más simple que el FTP.

Como la tasa de errores en las redes de área local suele ser muy baja, TFTP se apoya sobre UDP en vez de TCP, como ocurría con FTP. Para evitar la posibilidad de que se altere el orden de los mensajes, se retrasa el envío de un mensaje o bloque de datos hasta haber recibido confirmación del bloque anterior o hasta el vencimiento de n cronómetro de tiempo de espera máximo. Considerando la velocidad de las redes de área local, éste es un buen método.

TFTP solo maneja cuatro tipo de mensajes sobre la red:

- **Solicitud de lectura.** Enviado por un cliente para iniciar la lectura de un archivo desde el servidor.
- **Solicitud de escritura.** Enviado por un cliente para iniciar la escritura de un archivo en el servidor.
- **Bloque de datos.** Para comunicar un bloque de datos o mensaje del contenido total de un archivo. Tienen como máximo 512 bytes y llevan una cabecera con número de secuencia.
- **Confirmación.** Para confirmar la recepción de un bloque de datos.

21. Protocolos de aplicación sobre UDP: SNMP

El crecimiento de las redes en el seno de las empresas, y la diversidad de sistemas correspondientes, es decir routers de diversa procedencia, servidores centrales, de terminales, bridges locales y remotos, Hubs etc. imponen una gestión coherente del conjunto de toda esta estructura.

Vamos a dar, en este apartado, una breve descripción del estándar de gestión más difundido a través de todos los protocolos Internet: **SNMP.**

EL control de una red TCP/IP bajo SNMP reposa sobre **estaciones / puestos de control de red** (Los **«Managers»** o **manejadores**) que comunican con **elementos de red**, término que engloba cualquier equipo capaz de ejecutar TCP/IP: Servidores, Routers, terminales X, servidores de terminales, impresoras, repetidores, hubs, bridges ... y un largo etcétera de dispositivos físicos.

La porción de software que asume la gestión de la red sobre el elemento de red se denomina **agente.**

Las estaciones que explotarán la información almacenada por los **agentes,** son generalmente equipos de trabajo equipados con monitores en color y entornos gráficos. Presentan de forma intuitiva una variada información concerniente a los elementos de red o agentes.(Enlaces inactivos, volúmenes de tráfico etc.)

La comunicación puede ser bidireccional. **El manejador** solicita un valor específico al **agente** («¿Cuántos puertos ICMP inaccesibles ha generado Vd.?»), o bien el agente señala un evento importante al **manejador** («un interfaz ha perdido el enlace»). El **manejador** debe poder pedir al agente el valor de una variable y cambiarla («cambia el valor por defecto de IP TTL a 64»).

El control de una red TCP/IP descansa sobre 3 puntos:

1. Una **Base de información de gestión (MIB)** quien precisa cuáles son las variables soportadas por los elementos de la red. Información que podrá ser consultada o modificada por el **manejador**. La RFC 1213 define la segunda versión MIB-II.

2. Un juego de estructuras comunes y una nomenclatura utilizada para referenciar las variables dentro de la MIB, bajo el nombre de **Estructura de Gestión de Información** (SMI). Por ejemplo, SMI estipula que un «Counter» es un entero absoluto definido entre 0 y 4,294,967,295 que retorna a cero cuando alcanza el límite máximo.

3. El protocolo entre el **manejador** y el **agente**, el SNMP, está definido en la RFC 1157. Detalla el formato de los paquetes intercambiados. A pesar de soportar una larga serie de protocolos de transporte, UDP es el más utilizado.

Existen diversos productos comerciales que permiten la gestión SNMP a través de los MIB de todos aquellos dispositivos que lo soporten. Actualmente la mayoría de éstos disponen de SNMP. D-View y NMS son dos paquetes populares sobre plataforma WINDOWS que permiten efectuar gestión SNMP. Son productos de D-LINK y AMP respectivamente y disponen de grandes posibilidades en cuanto a flexibilidad de actuación y establecimiento de alarmas.

El significado y la catalogación de todas las variables del MIB sobrepasa el objetivo de este curso, existiendo documentación específica y de uso público sobre el tema. Nos limitare-

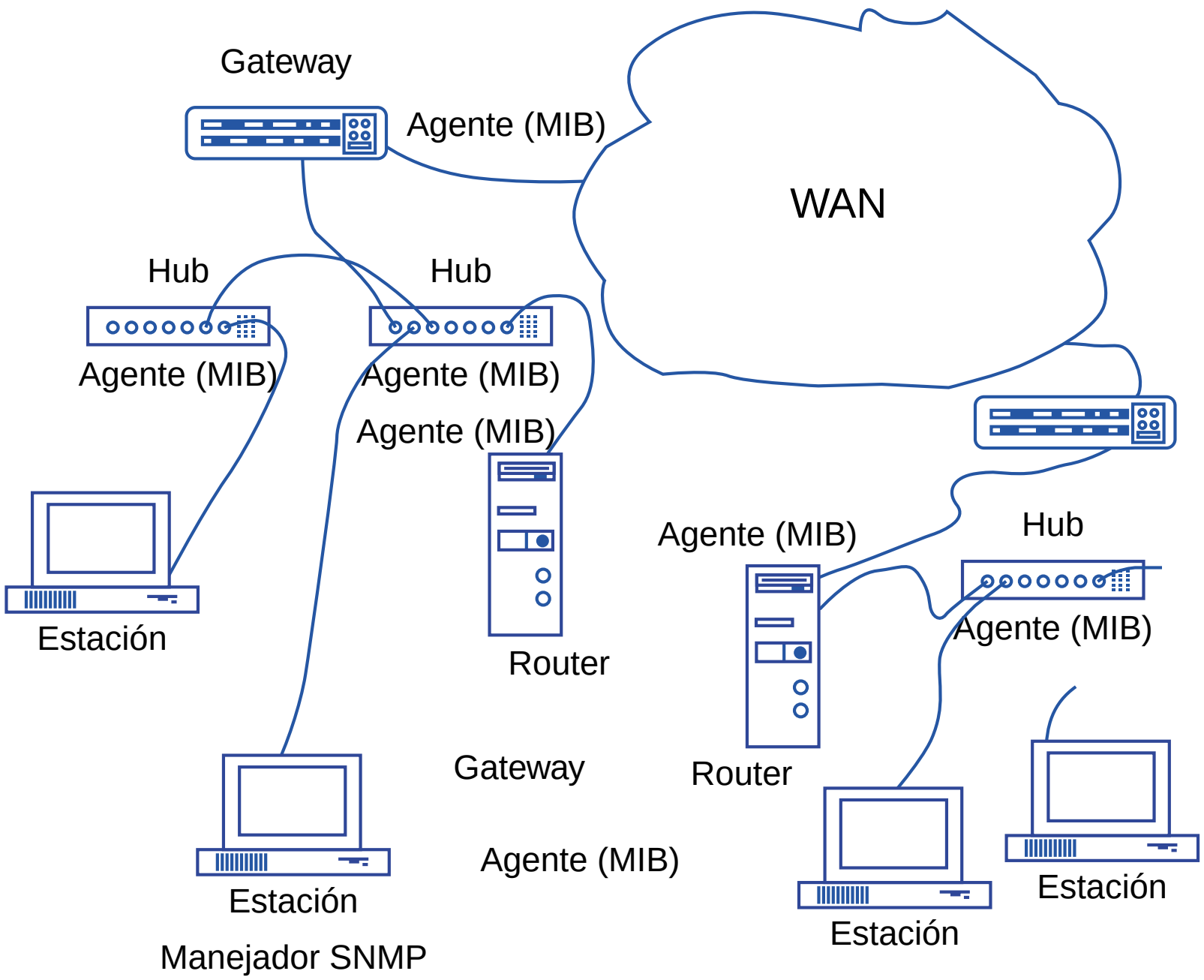


Figura 19

mos únicamente a utilizar algunos de los comandos que ofrece Unix al respecto.

El **daemon** utilizado se denomina **snmpd**, y es lanzado por la shell de arranque de tcp mediante el comando:

 # snmp start

El **manejador** envía sus peticiones (get-request, set-request o get-response) al puerto UDP 161. El **agente** envía **«traps»** o indicaciones de sucesos al puerto UDP 162. Gracias a la utilización de puertos distintos, un mismo sistema puede ser simultáneamente **manejador** y **agente**. Tal es el caso del servidor **std** o el **std2**.

22. Protocolos de aplicación sobre UDP: NFS

La mayor parte de la programación en red se efectúa escribiendo aplicaciones que realizan llamadas a funciones ofrecidas por el sistema para ejecutar operaciones específicas de red. Por ejemplo, una función realiza una apertura TCP activa, otra puede definir opciones específicas del protocolo, otra envía datos a través de un enlace TCP ya establecido, y así sucesivamente.

De forma general, el cliente envía comandos al servidor, y éste responde al cliente. Todas las aplicaciones utilizadas hasta ahora, Ping, routed, Telnet ...etc, están construidas de este modo.

Sin embargo, existe otra forma de realizar estas tareas. Las **RPC**, o llamadas de procedimientos distantes *(Remote Procedure Call)*, son otra forma de programar en red.

Un programa cliente realiza únicamente llamadas a subrutinas situadas en el programa servidor. Todos los detalles de la programación en red son ocultados por el software RPC.

Como ventajas podemos citar las siguientes:

1- La programación es más sencilla puesto que existe muy poca y en algunos casos nula programación asociada a la red. El programador de la aplicación escribe únicamente un programa cliente y las funciones del servidor llamadas por el cliente.

2- Si un protocolo sin corrección de errores como **UDP** es utilizado, detalles como el timeout y la retransmisión son gestionados por **RPC**.

3- La biblioteca **RPC** gestiona eventuales conversiones de datos para los argumentos y los valores de retorno. Es decir, si los argumentos consisten en números enteros de coma flotante, **RPC** se ocupa de todas las diferencias que pudieran existir en la forma de guardar los datos en el cliente y en el servidor. Ello simplifica la codificación de clientes y servidores que pudieran estar operando en entornos heterogéneos.

NFS (Network file system) proporciona un acceso transparente a los ficheros y a los sistemas de ficheros de un servidor. Accede únicamente a partes del fichero que referencia

un proceso, siendo uno de sus objetivos es el de proporcionar un acceso transparente. Significa que cualquier aplicación cliente que trabaje con un fichero local, podría trabajar con un fichero **NFS**, sin ninguna modificación del programa, cualquiera que sea.

NFS es una aplicación cliente-servidor basada en **Sun RPC**. Los clientes **NFS** acceden a los ficheros del servidor NFS enviando peticiones **RPC**.

22.1 Configuración de NFS desde TUN (MSDOS)

Entrando desde el directorio C:\TUNTCP y ejecutando TUNTCP, se selecciona la opción **«Network File System»** o sistema de ficheros de red.

Aparece un submenú con una serie de opciones, de las que seleccionamos «Nfs, parámetros iniciales». Seleccionamos la configuración «Standar» que nos proporcionará 2 discos virtuales. (Depende también de la cantidad de memoria de que dispongamos). Salimos salvando con F2 y regresamos al menú anterior.

Seleccionamos ahora el apartado «Setup - Sistema de Ficheros». Aquí tenemos que especificar el **Identificador de Volumen** de la unidad de disco virtual (por ejemplo

«std_nfs»), su **nombre** (por ejemplo D:) utilizando una letra de unidad no empleada y asegurándose de que el parámetro *LASTDRIVE* del *config.sys* está bien dimensionado, el nombre del **servidor** sobre el que queremos montarlo (por ejemplo std), el **directorio** destinado a *nfs* por el servidor y nuestro **nombre de usuario**.

Esta operación se realizará por cada unidad designada. Sin olvidar pulsar F2 para salvar los cambios, regresamos mediante «Escape» al símbolo del sistema.

A partir de aquí, lanzaremos en primer lugar la aplicación cliente, mediante el fichero **autonfs.bat.**

Si no aparece ningún error, estamos en condiciones de montar las unidades virtuales.

C:\TUNTCP\mount std_nfs

Disponemos ahora de una unidad de disco D: en nuestro ejemplo

22.2 Configuración de NFS desde UNIX

Los ficheros necesarios son 2 básicamente: **/etc/exports** y **/etc/hosts.** En este último, figura una relación entre los nombres de las estaciones y sus direcciones IP. Con el comando:

$more /etc/hosts

Se puede visualizar el contenido de esta tabla.

El fichero /etc/exports está directamente relacionado con la accesibilidad de los usuarios hacia el sistema. Su contenido podría ser algo similar a:

```
 /usr -access=clientes#exporta sólo a clientes

/usr/local              #exporta hacia todo el mundo

/usr2 -access=hermes:zip:tutorial   #exporta solamente a estas máquinas

/usr/sun      -root=hermes:zip  #proporciona privilegios de root a la máquina hermes

/usr/new      -anon=0 #proporciona a todas las máquinas privilegios de root

/usr/bin      -ro   #exporta en sólo lectura para todo el mundo

/usr/stuff    -access=zip,anon=-3,ro #pueden existir varias opciones en una sola línea
```

Se pueden ver los distintos modificadores y sus efectos sobre el acceso de los usuarios, cuyos nombres deben de haber sido introducidos previamente en el fichero **/etc/hosts.**

Una vez introducidos en el fichero **exports** los nombres de los directorios que deseamos exportar, debemos invocar el comando:

exportfs -a

De este modo el «daemon» nfs actualizará su tabla con los nuevos directorios exportados. El comando:

exportfs

Nos muestra los directorios actualmente exportados.

22.3 Unix - Unix NFS

Cada servidor dispone además, del servicio **NFS cliente** permitiendo ser ejecutado desde un servidor hacia el otro. Ello significa que podríamos tener dentro del servidor **std2** un subdirectorio de **std**, de forma totalmente transparente.

A modo de ejemplo vamos a suponer que quisiéramos montar un directorio **nfs** sobre la máquina **std2**, en un directorio vacío llamado /u0/Unfs. El directorio origen de la máquina **std** es **/u0/nfs1,** que contiene una serie de datos susceptibles de ser copiados en **std2.**

Primeramente deberíamos añadir una línea en el fichero **/etc/exports** de **std** conteniendo **/u0/nfs1 -anon=0.**

A continuación ejecutar el comando **«exportfs -a»**, para exportar las modificaciones realizadas sobre **exports.**

El comando ejecutado en **std2**, que nos permitirá realizar el montaje es:

 # mount -v -f NFS std:/u0/nfs1 /u0/Unfs

Indica que deseamos montar el filesystem **/u0/nfs1** de la máquina **std** (std:/u0/nfs1), sobre el subdirectorio **/u0/Unfs** de la máquina actual **(std2)**.

Una vez realizada la operación, podemos realizar una copia de los datos en /u0/nfs1 simplemente con:

 # cp /u0/Unfs/* /u1/datos

Suponiendo que /u1/datos fuera el destino de la copia.

Para deshacer la operación, el comando relacionado es:

 # umount /u0/Unfs

23. Cuestionario 3

1º) ¿Bajo qué tipo de programación se encuentra implementado el servicio NFS?

2º) Se disponen de dos máquinas Unix unidas por una conexión en fibra óptica, denominadas «fobos» y «deimos» respectivamente.

Describir la secuencia de pasos y comandos necesaria para montar el directorio /u0/disco de la máquina «fobos», sobre el directorio /u0/tmp de «deimos».

3º) ¿ Qué se necesita hacer para que el directorio /u0/disco de «fobos» sea exportable únicamente para la máquina «deimos»?

4º) ¿ Cuántos sockets consume un cliente en una conexión FTP?

23. Cuestionario 3

5º) Se desea realizar una transferencia de archivos entre dos sistemas Unix conectados a la Internet, «ganímedes» y «europa», desde un tercero denominado «io».

Descríbase algún modo de hacerlo.

24. Anexo A

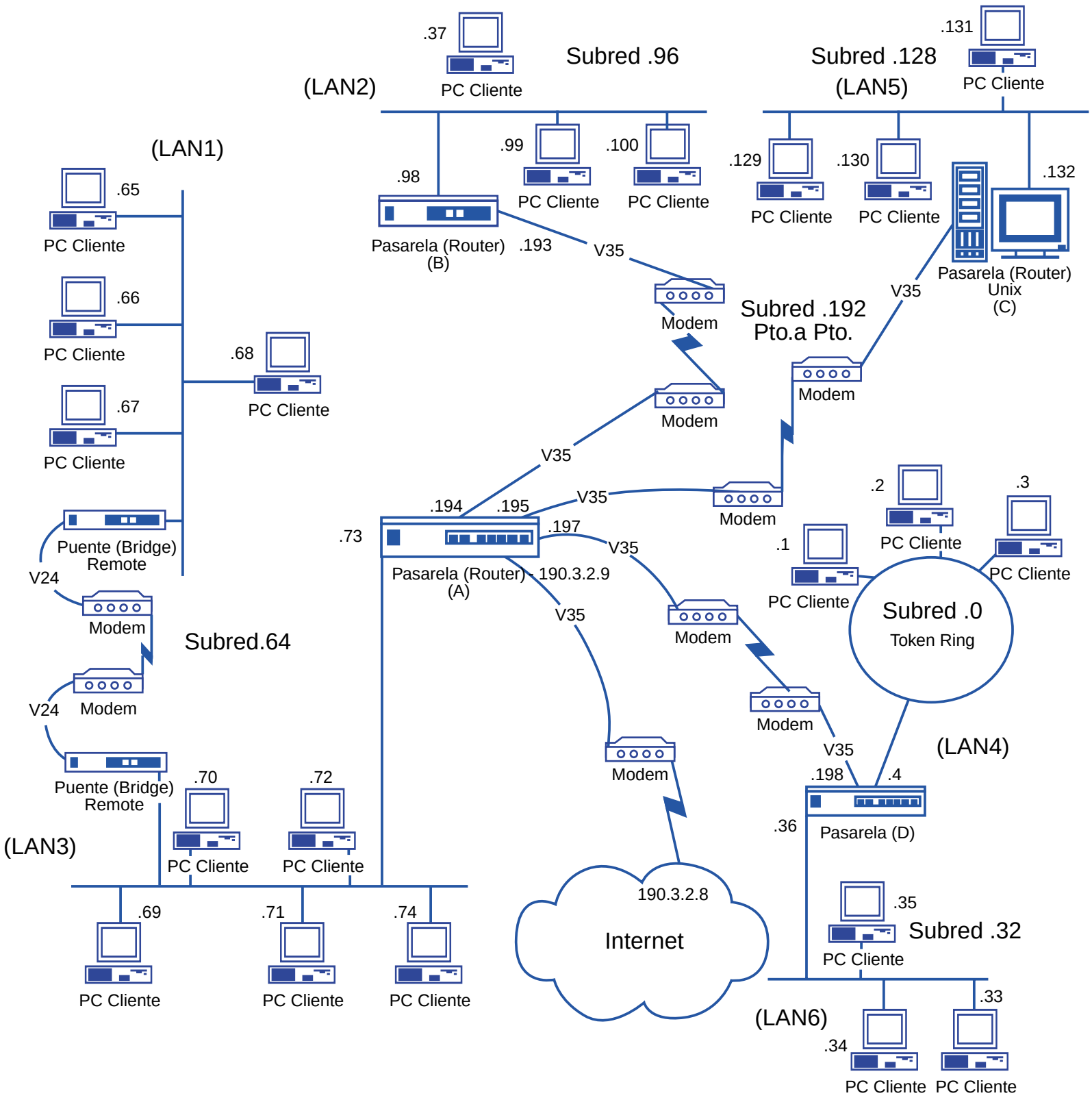
Red: 196.14.162.0

Máscara para todos los interfaces: 255.255.255.224 (3 bits)

Los Bridges conectan redes sólo a nivel de Enlace. Son transparentes al protocolo IP

Las conexiones punto a punto se agrupan en una única subred (.192)

24. Anexo A


(LAN1)
.65
PC Cliente
.66
PC Cliente
.67
PC Cliente
.68
PC Cliente
Puente (Bridge)
Remote
V24
Modem
Subred.64
V24
Modem
Puente (Bridge)
Remote
(LAN3)
.70
PC Cliente
.72
PC Cliente
.69
PC Cliente
.71
PC Cliente
.74
PC Cliente
.37
PC Cliente
(LAN2)
Subred .96
.98
.99
PC Cliente
.100
PC Cliente
Pasarela (Router)
(B)
.193
V35
Modem
Modem
V35
.73
.194
.195
.197
Pasarela (Router)
(A)
190.3.2.9
V35
V35
V35
Modem
Modem
Modem
V35
190.3.2.8
Internet
.131
PC Cliente
Subred .128
(LAN5)
.129
PC Cliente
.130
PC Cliente
.132
Pasarela (Router)
Unix
(C)
V35
Subred .192
Pto.a Pto.
Modem
.2
PC Cliente
.3
PC Cliente
.1
PC Cliente
Subred .0
Token Ring
(LAN4)
.198
.4
.36
Pasarela (D)
.35
PC Cliente
Subred .32
.33
(LAN6)
.34
PC Cliente
PC Cliente

25. Glosario de términos

10Base2

ACK

AM

ARP

Asentimiento Positivo con Retransmisión

AUI

BER

BNC

Broadcast

Cabecera IP

conexiones punto a punto

CRC

Creación de Rutas

datagrama IP

direcciones IP

25. Glosario de términos

Direcciones MAC

EGP

Encaminamiento Dinámico

Ethernet

ETHTCP.EXE

fabricantres

FIN

fragmentación

gated

HELLO

hostname

HUB

IEEE802.3

ifconfig

Interface de bucle local

Máscara de Subred

mensajes ICMP

MTU

netconfig

netstat

NFS

Número de asentimiento

Número de Puerto

25. Glosario de términos

Tabla de hosts

TCP

Telnet

TOS (Type of service)

tramas Ethernet

Transceptor

TTL (Time to Live)

TUNTCP.EXE

UDP

UTP

ventana

26. Bibliografía

W.R. Stevens
TCP/IP Illustrated: The Protocols
Addison-Wesley

Craig Hunt
TCP/IP Network Administration
O'Reilly & Associates, Inc.

Axis & Agix
CLIENT-SERVEUR. Programmation TCP/IP
Editions Laser

Douglas Comer
TCP/IP
InterEditions